Inverter

i700

E70ACM...

i700 servo inverter

Reference manual EN

13410788

Lenze

Contents

Contents

1 About this documentation

Danger!

The controller is a source of danger which may lead to death or severe injury of persons.

To protect yourself and others against these dangers, observe the safety instructions before switching on the controller.

Please read the safety instructions in the **mounting instructions** and in the **hardware manual** for the i700 servo inverter. Both instructions are included in the scope of supply of the drive controller.

Target group

This documentation addresses to all persons who want to parameterise, configure, and diagnose the i700 servo inverter.

Validity

The information in this documentation are valid for the following standard devices:

Product series	Type designation	From software version
i700 servo inverter	E70ACM...	01.00

Screenshots/application examples

All screenshots in this documentation are application examples. Depending on the firmware version of the i700 servo inverter and the software version of the engineering tools installed (»PLC Designer« or »EASY Starter«), the screenshots in this documentation may deviate from the screen representation.

Document history

Version			Description
1.5	03/2014	TD05	Extensions & corrections; Adaptation to software version V01.06
1.4	10/2013	TD05	Extensions & corrections; Adaptation to software versions V01.04 and V01.05
1.3	03/2013	TD05	Extensions & corrections; Adaptation to software version V01.03
1.2	10/2012	TD05	Extensions & corrections; Adaptation to software version V01.02
1.1	07/2012	TD05	First edition

Tip!

Information and tools regarding the Lenze products can be found in the Internet:

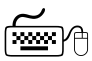
http://www.lenze.com → Download

1.1 Conventions used

This documentation uses the following conventions to distinguish between different types of information:

Type of information	Writing	Examples/notes
Spelling of numbers		
Decimal separator	Point	The decimal point is always used. Example: 1234.56
Hexadecimal number	0x	For hexadecimal numbers, the prefix "0x" is used. Example: 0x60F4
Binary number	0b	For binary numbers, the prefix "0b" is used. Example: 0b00010111
Text		
Version information	Blue text colour	All information that only applies to a certain controller software version or higher is identified accordingly in this documentation. Example: This function extension is available from software version V3.0!
Program name	» «	The Lenze »PLC Designer« PC software...
Window	*italics*	The *Message* window ... / The *Options* dialog box...
Variable name		Set *bEnable* to TRUE to...
Control element	**bold**	The **OK** button... / The **Copy** command... / The **Properties** tab... / The **Name** input field...
Sequence of menu commands		If the execution of a function requires several commands, the individual commands are separated by an arrow: Select **File→Open** to...
Shortcut	**<bold>**	Press **<F1>** to open the online help.
		If a command requires a combination of keys, a "+" is placed between the key symbols: Use **<Shift>+<ESC>** to...

Program code	`Courier`	`IF var1 < var2 THEN`
		`a = a + 1`
		`END IF`

Keyword	**`Courier bold`**	
Hyperlink	underlined	Optically highlighted reference to another topic. In this documentation it is activated by mouse-click.
Icons		
Page reference	(📖 13)	Optically highlighted reference to another page. In this documentation it is activated by mouse-click.
Step-by-step instructions		Step-by-step instructions are indicated by a pictograph.

1.2 Terminology used

Term	Meaning
Engineering Tools	Software solutions for simple engineering at all stages
	»EASY Navigator« – Ensures easy operator guidance • All practical Lenze engineering tools at a glance • Tools can be selected quickly • Clearly arranged, simplifying the engineering process from the start
	»EASY Starter« – Simple tool for service technicians • Especially developed for the commissioning and maintenance of Lenze devices • Graphical user interface with few buttons • Simple online diagnostics, parameterisation and commissioning • No risk of accidentally changing the application • Ready applications can be loaded to the device
	»PLC Designer« – for process programming • Creation of individual programs • Programming of Logic & Motion in accordance with IEC 61131-3 (IL, LAD, FBD, ST, SFC, and CFC editor), is based on CoDeSys V3 • Certified function blocks according to PLCopen Part 1 + 2 • Graphical DIN 66025 editor (G-code) with DXF import • Integrated visualisation for an easy process representation • All important pieces of information at a glance during commissioning
Lenze Controller	The Lenze Controller (abbreviated: "Controller") is the central component of the automation system which (by means of the runtime software) controls the Logic and Motion functionalities. The Lenze Controller communicates with the field devices via the fieldbus.
Engineering PC	Use the Engineering PC and the engineering tools installed to configure and parameterise the system. The Engineering PC communicates with the Lenze Controller via Ethernet.
EtherCAT	EtherCAT® is a real-time capable Ethernet system with top performance. EtherCAT® is a registered trademark and patented technology licensed by Beckhoff Automation GmbH, Germany.
HIPERFACE®	HIPERFACE® stands for High Performance Interface and is a universal interface between motor feedback system and inverter. HIPERFACE® is a registered trademark of the SICK STEGMANN GmbH.
Object	"Container" for one or several parameters which can be used to parameterise or monitor the i700 servo inverter.
Index	For the purpose of addressing, each object is provided with a unique index. In this documentation the index is represented as a hexadecimal value and is identified by a prefixed "0x", e.g. "0x1000".
Subindex	If an object contains several parameters, the individual parameters are stored under "subindexes". In this documentation the colon is used as a separator between the index and the subindex, e.g. "0x1018:1".
Touch probe	A "Touch probe" is an event which can for instance be actuated in an edge-controlled manner via a digital input to detect an actual value (that changes quickly) at the time of activation and to process it further within the program afterwards.

1.3 Definition of notes used

The following signal words and symbols are used in this documentation to indicate dangers and important information:

Safety instructions

Layout of the safety instructions:

Danger!

(characterises the type and severity of danger)

Note

(describes the danger and gives information about how to prevent dangerous situations)

Pictograph	Signal word	Meaning
	Danger!	**Danger of personal injury through dangerous electrical voltage** Reference to an imminent danger that may result in death or serious personal injury if the corresponding measures are not taken.
	Danger!	**Danger of personal injury through a general source of danger** Reference to an imminent danger that may result in death or serious personal injury if the corresponding measures are not taken.
	Stop!	**Danger of property damage** Reference to a possible danger that may result in property damage if the corresponding measures are not taken.

Application notes

Pictograph	Signal word	Meaning
	Note!	Important note to ensure trouble-free operation
	Tip!	Useful tip for simple handling
		Reference to another document

1.4 Structure of the parameter descriptions

All parameters which you can use to parameterise or monitor the i700 servo inverter are stored within "objects".

- For the purpose of addressing, each object is provided with a unique index. In this documentation the index is represented as a hexadecimal value and is identified by a prefixed "0x", e.g. "0x1000".
- If an object contains several parameters, they are stored in "subindexes". In this documentation the colon is used as a separator between the index and the subindex, e.g. "0x1018:1".

Note!

This documentation is valid for the i700 servo inverter in the single axis version (single inverter) and also as double axis (double inverter).

For parameters referring to one axis, both indexes (for axis A and axis B) are listed in the parameter description. For a single axis, only the first index is relevant in this case.

Each parameter description is structured according to the following pattern:

Example: Structure of the parameter descriptions in this documentation

❶ ❷ ❸
0x2942 | 0x3142 - Current controller: Parameter

❹

Sub.	Name	Lenze setting	Data type
1	Current controller: Gain	148.21 V/A	UNSIGNED_32
2	Current controller: Reset time	3.77 ms	UNSIGNED_32

❺

Subindex 1: **Current controller: Gain**				
Setting range (min. value | unit | max. value)			Lenze setting	
0.00	V/A	750.00	148.21 V/A	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/100	UNSIGNED_32

❺

Subindex 2: **Current controller: Reset time**				
Setting range (min. value | unit | max. value)			Lenze setting	
0.01	ms	2000.00	3.77 ms	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/100	UNSIGNED_32

❶	Object index for axis A
❷	Object index for axis B (only relevant for double axis)
❸	Parameter or object name
❹	If the object contains several parameters: Overview table with list of all subindexes
❺	Table with detailed information about the corresponding parameter: • Explanations & references (optional) • Display options/possible settings, Lenze setting, attributes (for the meaning see the following table)

Parameter attributes

Name	Meaning	
Write access	☑ = Parameter can be written to. ☐ = Parameter can only be read.	
CINH	☑ = Parameter can only be written to if controller inhibit is set.	
OSC	☑ = Parameter can be recorded by means of the oscilloscope function.	
P	☑ = Parameter can be persisted.	
Tx	☑ = Parameter can be mapped into the TPDO.	
Rx	☑ = Parameter can be mapped into the RPDO.	
Data type	Data type of the parameter:	
	INTEGER_8	1 byte, with sign
	INTEGER_16	2 bytes with sign
	INTEGER_32	4 bytes with sign
	UNSIGNED_8	1 byte without sign
	UNSIGNED_16	2 bytes without sign
	UNSIGNED_32	4 bytes without sign
	UNSIGNED_64	8 bytes without sign
	STRING(xx)	ASCII string (with character length xx)
	ARRAY [] OF...	ARRAY
Scaling	Scaling of the parameter	

Tip!

To find a specific object or parameter in this documentation, you can use the following navigation helps:

- At the beginning of each main chapter, all objects which are described in the respective chapter are listed in a table.
- In the Table of attributes and the Index, all objects/parameters are listed with a reference to the detailed description.

2 Parameter handling

2.1 Parameter transfer during initialisation

During the acceleration of the system, the controller and the controllers exchange configuration data (e.g. bus cycle and PDO mapping). With regard to this, observe the following particular features for the i700 servo inverter:

Parameter download

The i700 servo inverter itself does not store parameter settings safe against mains failure. All settings deviating from the i700 servo inverter "Lenze setting" (default) are maintained centrally in the controller and are stored there permanently (persistently). During the initialisation at run-up, only these deviations are transferred to the i700 servo inverter by the controller. Like this it is ensured that the i700 servo inverter works with the parameter settings provided for it.

Firmware download (optional)

If required, the firmware of the i700 servo inverter can be stored together with the »PLC Designer« project. During the run-up, the controller then checks whether the firmware version in the i700 servo inverter complies with the firmware version stored in the project for this device. If this is not the case, the controller loads the firmware version stored in the project to the i700 servo inverter. Like this, it can be ensured for "Device replacement" service work that the replacement device also works with the same firmware version as the original device.

2.2 Storage parameter set (par001.*) and total parameter set (par000.*)

For the storage of the i700 servo inverter parameters, two different parameter sets are provided, which are stored in different parameter set files in the higher-level controller:

Storage parameter set (par001.i7psf)

- Only contains the parameters of the i700 servo inverter which are writable and identified with the "P" attribute (persistent).
- Parameters can be read out from the i700 servo inverter and saved to the file. Conversely, the parameters stored can be written to the i700 servo inverter again.
- Recommended for storage and archiving of the controllers settings.

Total parameter set (par000.i7psf)

- Contains all parameters of the i700 servo inverter, including the mere display parameters which vary permanently during operation.
- The total parameter set can only be read from the i700 servo inverter.
- Use for purposes of service and diagnostics.

The structure of the parameter set file is described in the appendix.

▸ Structure of the parameter set file (📖 310)

2.2.1 Saving a parameter set from the i700 to a file (export)

Reading out and storing the parameters from the i700 servo inverter is initiated and controlled from the higher-level controller. For reading out a parameter set, the corresponding parameter values are summarised in a parameter set file (par001.* or par000.*) in the i700 servo inverter and are then transferred as a file to the controller via EtherCAT.

2.2.2 Loading the stored parameter set to the i700 (import)

Loading a stored parameter set file to the i700 servo inverter is initiated and controlled from the higher-level controller, just like in the case of the read-out. The storage parameter set file is transferred to the i700 servo inverter as a file via EtherCAT, and the parameter settings are loaded (imported).

For logical reasons, it is only possible to transfer "storage parameter set files" (par001.*) to the i700 servo inverter. Display parameters are updated under normal operating conditions and do not need to be loaded.

Note!

In the case of Lenze controllers, the parameter set files are transferred as "files" via GCI ("Generic Communication Interface").

- GCI is a Lenze-specific application protocol which uses CoE ("CANopen over EtherCAT").
- Other control manufacturers may also use FoE ("File Access over EtherCAT") for the transfer.

2.2.3 Monitoring of the parameter import (error report)

When the parameter import has been completed, the "parErr.i7psf" error report file is created. This file is recreated automatically during every import.

On the basis of the error report it can be determined whether errors have occurred during the parameter import:

- Parameters that have been transferred with errors are listed there together with their index, subindex, and error code (SDO abort code).
- If no errors have occurred, this data area of the file is blank. However, the headers including the checksum are available and valid nevertheless.

2.3 Cyclic redundancy check (CRC) - parameter set comparison on the basis of the checksum

Each parameter set features an individual checksum (CRC32), which is composed of the settings of all storable indexes (P-flag) of the parameter set. When a setting changes, there is also a change in the checksum.

Cyclic redundancy check (CRC):

By comparison of the checksums of parameter sets it can be determined very quickly whether parameter sets are identical or not. In particular, it is thus determined for the i700 servo inverter whether the parameter set stored in the controller is identical to that in the i700 servo inverter and therefore does not have to be loaded again.

0x2030 - Parameter set: Validity check (CRC)

Checksum for quick comparison of the storage parameter set (par**001**.i7psf)
- If the checksum shown here is identical to the checksum of the parameter set file stored in the Lenze Controller, it is not necessary to write the parameter set file to the i700 servo inverter.
- If the checksums are different, however, there are deviations between the parameter set available in the Lenze Controller and that in the i700 servo inverter.
- The checksum cannot be used for comparison of the total parameter set (par**000**.i7psf).

Display area (min. value | unit | max. value)			Initialisation	
0		4294967295		
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX				UNSIGNED_32

3 Communication with the controller

"EtherCAT control technology" communication manual

Here you will find detailed information on the EtherCAT configuration and commissioning of Lenze devices in the EtherCAT network.

Objects described in this chapter

Object		Name	Data type
Axis A	Axis B		
0x2020		EoE information	RECORD
0x2824	0x3024	Device control via PDO: Activation	UNSIGNED_8
0x2830	0x3030	Lenze control word	UNSIGNED_16
0x2831	0x3031	Lenze status word	UNSIGNED_16
0x2833	0x3033	Lenze statusword 2	UNSIGNED_16

Tip!

The communication objects important for the integration of the controller with external control are described in the appendix.

▶ Communication objects (🕮 313)

3.1 Acceleration of the system (initialisation)

Stop!

Before switching on the i700 servo inverter for the first time, check the entire wiring with regard to completeness, short circuit, and earth fault.

To establish communication, the i700 servo inverter must be supplied with voltage. During the acceleration of the system, the controller and the drive exchange configuration data.

The controller transmits the following configuration data to the i700 servo inverter:

- Bus cycle
  - This is the basic cycle within which the EtherCAT bus is actuated. The bus cycle is given as a multiple of 125 µs.
  - The bus cycle equals the communication cycle within which the process data are exchanged cyclically. New process data are only accepted and generated in the Servo-Inverter i700 maximally every 250 µs.
- Parameter set determined during commissioning.
  - Among other things, it includes information about feedforward control values, the mains voltage and the switching frequency as well as controller parameters adapted to the motor module which are used for the motor control.
- Configuration of the process data transmitted cyclically via EtherCAT (PDO mapping)

Note!

If no data for the initialisation of the controller are transmitted, the i700 servo inverter uses the "Lenze setting" for the parameters.

Power up/Power down

In the case of "Power up" and "Power down", no undefined states can occur that cause damage to the device or motor movements which are not requested or not braked.

In the case of a voltage failure/dip, there is no immediate response by the i700 servo inverter. The functionality is to be maintained for as long as possible.

▸ 24-V supply voltage monitoring (📖 235)

Persistent data storage

A complete, persistent storage of the drive configuration which remains the same even though the controller is switched off from time to time is carried out in the higher-level controller instead of in the i700 servo inverter. Hence, the controller needs to transmit the configuration data again to the i700 servo inverter after a power up.

However, in the case of a power down, the following device data are stored persistently in the i700 servo inverter:

- Power-on and elapsed-hour meter (0x2D81 or 0x3581 for axis B)
- History buffer (📖 265)

3.2 Process data (cyclic PDO transfer) and PDO mapping

"EtherCAT control technology" communication manual

Here you will find some detailed information on the configuration of the process data objects (PDO mapping) with the »PLC Designer«.

Cyclic process data are transferred cyclically between the controller (master) and the controllers (slaves) as so-called *Process Data Objects* (PDOs).

- The i700 servo inverter supports the following bus cycle times of the EtherCAT:
  0.125 ms*
  0.250 ms
  0.500 ms
  1.000 ms
  ... (only integer multiple of 1 ms)
  10.000 ms (max. cycle time)
  * **Note:** With a bus cycle time of 0.125 ms, the process data is updated only every 0.250 ms, as this is the fastest control cycle for the setpoint and actual value transfer.
- The processing time of a process date through the drive is t=max[bus cycle, 0.250 ms]. If the entire chain from the control via the drive back to the control is considered, a process data needs 2 bus cycles in addition (for Lenze C3200) . Thus, the turnaround time for a PDO can be indicated with 3 bus cycles.
- For the process data communication, the i700 servo inverter supports the mapping of max. 32 process data objects (PDOs) with a total max. size of 100 bytes per direction of transmission.
- A fixed PDO mapping preconfigured by Lenze is available for every CiA402 operating mode supported by the i700 servo inverter. Every PDO mapping includes of several objects from the Object directory.
- For every axis, further fixed, preconfigured PDO mappings can be used for touch probe functionality.
- In addition to the fixed, preconfigured PDO mappings, freely configurable PDO mappings are available for every axis which can be used for individual PDO mapping. A maximum of 8 objects from the Object directory can be configured per direction of transmission.
- The configuration of the PDOs actually transmitted between the controller and the i700 servo inverter is carried out via the »PLC Designer«.
  - Experience shows that most i700 servo inverters are operated in one of the available CiA402 operating modes (csp, csv, cst or vl). We therefore recommend to use and activate the fixed PDO mapping preconfigured by Lenze for the selected operating mode. In doing so all parameters that usually need to be replaced during a cyclic PDO transfer can be accessed.
  - If you wish to use touch probe functionality in addition, optionally and additionally activate the fixed, preconfigured PDO mapping for touch probe functionality.
  - Moreover, the PDO mappings that can be freely configured by the user can be optionally activated. However, we recommend to use these freely configurable PDO mappings in special cases only, if no standard PDO mapping is suitable for the case of application at hand. Special thought should be given to the aspect of traceability in case of service.

Object		Info
Axis A	Axis B	
RPDO mapping – configuration of the process data (setpoints) from the controller to the i700 servo inverter		
0x1600	0x1610	Fixed, preconfigured PDO mapping object for "Cyclic sync position mode (csp)"
0x1601	0x1611	Fixed, preconfigured PDO mapping object for "Cyclic sync torque mode (cst)"
0x1602	0x1612	Fixed, preconfigured PDO mapping object for "Cyclic sync velocity mode (csv)"
0x1603	0x1613	Fixed, preconfigured PDO mapping object for "Velocity mode (vl)"
0x1604	0x1614	Fixed, preconfigured PDO mapping object for "Touch probe (TP)"
0x1605	0x1615	Freely configurable PDO mapping object
0x1606	0x1616	Fixed, preconfigured PDO mapping object for torque limit values
TPDO mapping – configuration of process data (actual values) from the i700 servo inverter to the controller		
0x1A00	0x1A10	Fixed, preconfigured PDO mapping object for "Cyclic sync position mode (csp)"
0x1A01	0x1A11	Fixed, preconfigured PDO mapping object for "Cyclic sync torque mode (cst)"
0x1A02	0x1A12	Fixed, preconfigured PDO mapping object for "Cyclic sync velocity mode (csv)"
0x1A03	0x1A13	Fixed, preconfigured PDO mapping object for "Velocity mode (vl)"
0x1A04	0x1A14	Fixed, preconfigured PDO mapping object for "Touch probe (TP)"
0x1A05	0x1A15	Freely configurable PDO mapping object

3.2.1 Synchronisation with "Distributed clocks" (DC)

The "Distributed clocks" (DC) function enables an exact time adjustment for applications where several auxiliary axes carry out a coordinated movement at the same time. The data is accepted synchronously with the PLC program. In the case of the DC synchronisation, all slaves are synchronised with a reference clock, called the "DC master".

"EtherCAT control technology" communication manual

Here you will find some detailed information about DC synchronisation.

3.3 Parameter data transfer (SDO communication)

In addition to the cyclic process data transfer, parameter data can be transferred as so-called SDOs (Service Data Objects) in a non-cyclic manner within an individual datagram between the controller (master) and the controllers (slaves).

- SDO communication is implemented according to the EtherCAT-CoE protocol, using a mailbox.
- SDO communication enables read or write access to all indices contained in the object directory of the i700 servo inverter.
- The turnaround time for an SDO (request by the control, transport via the bus, processing in the drive, transport of the response back to the control) is between 1 ms and 100 ms plus 2 * bus cycle time (typically 10 ms).

3.3.1 Object directory

The object directory contains the specific indices for all axes. The object directory is structured according to specifications of the EtherCAT Technology Group (ETG):

Range	Index area		
	Device	Axis A	Axis B
Communication Area	**0x1000 - 0x1FFF**		
Identification data	0x1000 - 0x1018	-	-
Sync manager	0x1C00 - 0x1C33	-	-
PDO mapping	-	0x1600 - 0x1606 0x1A00 - 0x1A05	0x1610 - 0x1616 0x1A10 - 0x1A15
Manufacturer specific area	**0x2000 - 0x5FFF**		
Device settings	0x2000 - 0x27FF	-	-
Axis identification	-	0x2800 - 0x281F	0x3000 - 0x301F
Axis control	-	0x2820 - 0x283F	0x3020 - 0x303F
Error management	-	0x2840 - 0x28FF	0x3040 - 0x30FF
Motor control & motor settings	-	0x2900 - 0x2CFF	0x3100 - 0x34FF
Touch probe	-	0x2D00 - 0x2D3F	0x3500 - 0x353F
Monitoring functions	-	0x2D40 - 0x2D7F	0x3540 - 0x357F
Diagnostics	-	0x2D80 - 0x2DBF	0x3580 - 0x35BF
Service/internal	-	0x2DC0 - 0x2E3F	0x35C0 - 0x363F
Reserved	-	0x2E40 - 0x2FFF	0x3640 - 0x37FF
CiA402 profile specific area	**0x6000 - 0xDFFF**		
Device profile CiA402	-	0x6000 - 0x67FF	0x6800 - 0x6FFF

3.3.2 SDO abort codes

If an SDO request is evaluated negatively, a corresponding abort code is output:

SDO abort code	Description
0x0000 0000	No error
0x0503 0000	The status of the toggle bit has not changed.
0x0504 0000	SDO protocol time-out
0x0504 0005	The space in the main memory is not sufficient.
0x0601 0000	Access to object not supported.
0x0601 0001	Read access to a write-protected object.
0x0601 0002	Write access to a write-protected object.
0x0602 0000	Object is not listed in the object directory.
0x0604 0041	Object cannot be mapped into the PDO.
0x0604 0042	The number and/or length of the mapped objects would exceed the PDO length.
0x0604 0043	General parameter incompatibility
0x0604 0047	General internal device incompatibility
0x0606 0000	Access has failed because of hardware errors.
0x0607 0010	Wrong data type or parameter length.
0x0607 0012	Wrong data type (parameter length is too large).
0x0607 0013	Wrong data type (parameter length is too small).
0x0609 0011	Subindex does not exist.
0x0609 0030	The value range for parameters is too large (only for write access).
0x0609 0031	The parameter value is too high.
0x0609 0032	The parameter value is too low.
0x0800 0000	General error
0x0800 0020	Data cannot be transferred or saved to the application.
0x0800 0021	Data cannot be transferred or saved to the application because of local control.
0x0800 0022	Data cannot be transferred/saved to the application because of current device state.

3.3.3 ESI: EtherCAT Slave Information file (device description)

The EtherCAT Slave Information file (EtherCAT Device Description file) contains all information about the device (operating modes, parameters, ...).

- The EtherCAT Slave Information file is integrated by the EtherCAT network configuration tool in order to be able to configure and commission the devices.
- Part of the information contained in the EtherCAT Slave Information file can be uploaded online by the EtherCAT master by accessing the EtherCAT EEPROM of the device. The description of the object directory can also be identified online.

3.4 Activating the control via PDO

0x2824 | 0x3024 - Device control via PDO: Activation

This object serves to switch off all RPDOs (from the device's point of view) so that the device is exclusively controlled via SDOs.

- This is, for instance, required for manual enable of commissioning functions and test modes via the (0x6040 control word 0x6840 for axis B). ▸ Enable/inhibit via control word

Selection list (Lenze setting printed in bold)			
0	Off		
1	**Activate**		
☑ Write access ☑ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_8

3.5 Lenze control and status word

0x2830 | 0x3030 - Lenze control word

Via the Lenze control word, the control functions can be influenced.

Setting range (min. value | unit | max. value)			Lenze setting
0x0000		0xFFFF	0x0000
Value is bit-coded: (☑ = bit set)			Info
Bit 0 ☐	Flying restart: Completed		Via this bit, the control reports the acceptance of the speed found to the "Flying restart" function. The flying restart process is now completed.
Bit 1 ☐	Flying restart: Blocked		"1" ≡ Block flying restart process
Bit 2 ☐	Reserved		
Bit 3 ☐	Reserved		
Bit 4 ☐	Speed controller: Load I component		"1" ≡ Set starting value of the torque • In case of servo control, this corresponds to the I component of the speed controller, in case of V/f operation to the modulation of the slip compensation. • As long as this bit is set to "1", the I component and the slip compensation are set to the starting value set in 0x2902 (or 0x3102 for axis B).
Bit 5 ☐	Position: Traverse to new actual position		"1" ≡ Set/relatively shift actual position • Axis A: Set the actual position (0x6064) under consideration of the set resolution (0x608F) to the value set in 0x2983 (0x2984 = 0), or shift it by the value set in 0x2983 (0x2984 = 1). • Axis B: Set the actual position (0x6864) under consideration of the set resolution (0x688F) to the value set in 0x3183 (0x3184 = 0), or shift it by the value set in 0x3183 (0x3184 = 1).
Bit 6 ☐	Activate DC-injection braking or short-circuit braking		"1" ≡ Trigger DC-injection braking for asynchronous motor or short-circuit braking for synchronous motor
Bits 7-15 ☐	Reserved		
☑ Write access ☐ CINH ☑ OSC ☐ P ☑ RX ☐ TX			UNSIGNED_16

0x2831 | 0x3031 - Lenze status word

In the Lenze status word, messages are combined that go beyond the CiA specification.

Display area (min. value | unit | max. value)			Initialisation
0x0000		0xFFFF	
Value is bit-coded:			Info
Bit 0	Position controller: In limitation		Position mode: Output of the position controller in limitation
Bit 1	Speed: Limited speed setpoint 1		Input of speed controller 1 in limitation
Bit 2	Speed controller: In limitation		Output of speed controller 1 in limitation
Bit 3	Torque: Limited target torque		Target torque in limitation
Bit 4	Motor: Limited current setpoint		Setpoint current in limitation
Bit 5	Speed: Limited speed setpoint 2		Torque mode: Input of speed controller 2 in limitation
Bit 6	Upper speed limit is active		Torque mode: Speed is limited to the upper speed limit (0x2946:1 or 0x3146:1 for axis B)
Bit 7	Lower speed limit is active		Torque mode: Speed is limited to the lower speed limit (0x2946:2 or 0x3146:2 for axis B)
Bit 8	Flying restart in progress ...		V/f operation: "Flying restart process" function is active
Bit 9	Flying restart: Ready for operation		V/f operation: "Flying restart process" function has acquired speed
Bit 10	Limited output frequency		V/f operation: Setpoint frequency in limitation
Bit 11	Asynchronous motor magnetised		In case of servo control for synchronous motor (SM): • Bit is always set. In case of servo control for asynchronous motor (ASM): • Bit is set if the difference between setpoint and actual flux is smaller than 10 % of setpoint flux. • Bit is reset if the difference has increased to 15 % of setpoint flux. In case of V/f operation: • Bit is set if the rotor time constant has been passed seven times, calculated from the time the controller has been enabled and no restart on the fly has been active and the total motor current has reached 20 % of the rated motor current for the first time. Otherwise 0.
Bit 12	Motor phase failure detection in progress ...		Motor phase failure detection is active
Bit 13	Feedback: Open circuit		Position feedback is interrupted
Bit 14	Delay time: 'Reset error' is active		The error cannot be reset before the delay time has expired. The remaining delay time is displayed in 0x2840 (or 0x3040 for axis B).
Bit 15	Clamp is active		V/f operation: Clamp (short-time inhibit of the inverter) is active
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			UNSIGNED_16

0x2833 | 0x3033 - Lenze statusword 2

From version 01.03

Display area (min. value | unit | max. value)			Initialisation
0x0000		0xFFFF	
Value is bit-coded:			Info
Bit 0	Feedback modification		Changes in the settings of the feedback system have been executed. The status bit is reset at controller enable.
Bit 1	Manual test mode active		A manual test mode is active (0x2825 or 0x3025 = 1, 2 or 3)
Bit 2	Manual control mode active		Manual control mode is active (0x2825 or 0x3025 = 4)
Bit 3	Angle tracking control: Active		Reserved
Bit 4	Absolute value encoder selected		From version 01.06 Absolute feedback system selected (parameterised) • In case of the "resolver" device variant, this bit is set if the resolver pole pair number "1" is set in 0x2C43 (or 0x3443 for axis B). • In case of the "encoder" device variant, this bit is set if the encoder type "2: Hiperface absolute value encoder" is set in 0x2C40 (or 0x3440 for axis B).
Bit 5	Absolut position available		From version 01.06 Initialisation of the selected feedback system is completed. All information of the feedback system have been transmitted. • In case of an error in the feedback system, this bit is set to "0" as long as the error exists.
Bit 6	DC-injection braking or short-circuit braking: Active		From version 01.06 DC-injection braking for asynchronous motor or short-circuit braking for synchronous motor is active
Bit 7-15	Reserved		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			UNSIGNED_16

3.6 Ethernet over EtherCAT (EoE)

From version 01.05, the i700 servo inverter supports the "Ethernet over EtherCAT (EoE)" protocol.

The "Ethernet over EtherCAT (EoE)" protocol serves to send standard Ethernet telegrams via the EtherCAT network without the real-time communication of the EtherCAT process data being affected.

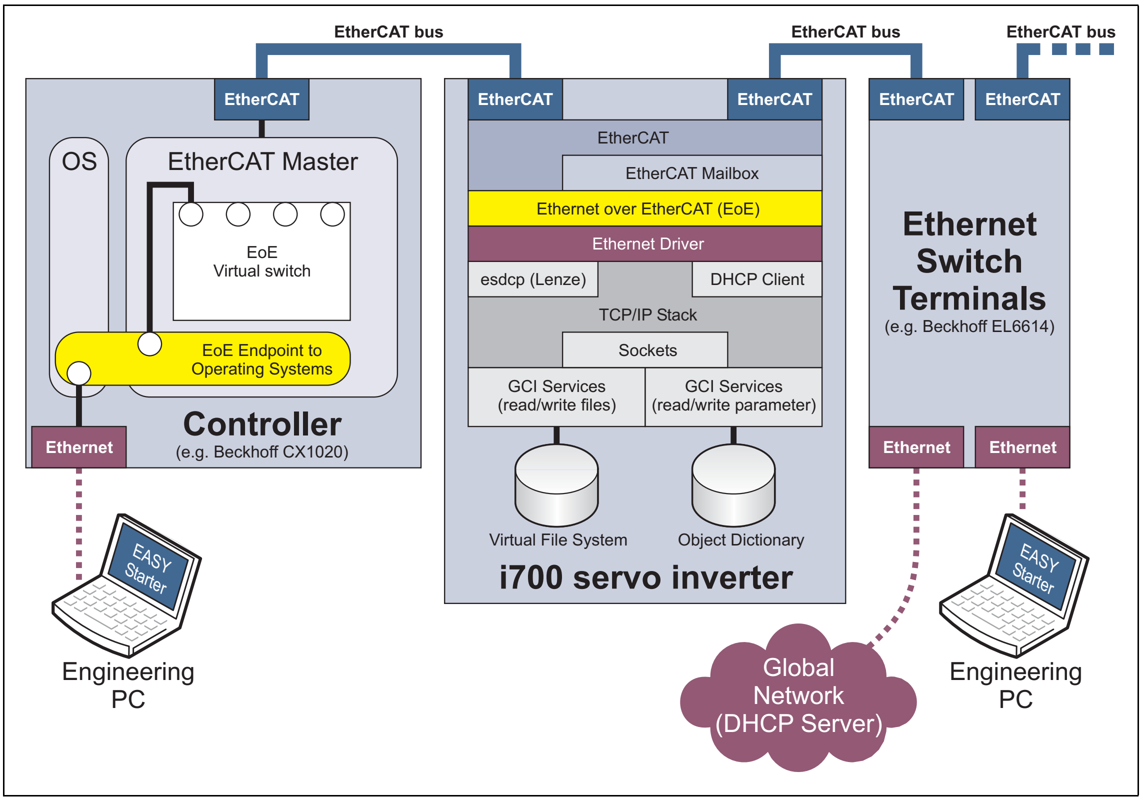
This extension allows for establishing a parameter communication (SDO communication) with the Servo-Inverters i700 on the EtherCAT bus via a standard Ethernet connection (e.g.from a PC with »EASY Starter«).

3.6.1 System architecture

The following illustration shows the system architecture for an EtherCAT network with EoE nodes. This network provides access per PC with »EASY Starter« via two interfaces:

A. Access via an Ethernet-Switchport terminal (e.g. Beckhoff EL6614)

B. Access via control (e.g. Beckhoff CX1020)
For this purpose, the control has to support IP routing and the "EoE Endpoint to Operating Systems" functionality according to the "ETG.1500 master classes" specification.



[3-1] System architecture for an EtherCAT network with EoE nodes

The IP address allocation in the network can both be made statically by the EtherCAT master via configuration and via a higher-level infrastructure which can be connected to the Ethernet Switchport terminal.

The PC used for diagnostic purposes can get the IP address via the Ethernet Switchport terminal if this supports a DHCP server or BOOTP mechanism or it must feature a statically allocated IP itself in the same subnetwork.

3.6.2 Supported protocols and services

- ARP
- DHCP Client
- ESDCP
- ICMP (ping)
- IP
- UDP/TCP
- GCI-SDO communication (TCP port 9410)

3.6.3 Display of EoE-specific information

The following object serves to read out the EoE-specific information for diagnostic purposes.

0x2020 - EoE information

From version 01.05
- The subcodes 1 ... 6 display the IP and Ethernet settings configured by the control during initialisation.
- The subcodes 7 and 8 display the traffic via EoE in Rx and Tx direction. The displayed values can be reset to "0" by entering any value.

Sub.	Name	Lenze setting	Data type
1	Virtual MAC address		STRING(32)
2	IP address		STRING(32)
3	Subnet mask		STRING(32)
4	Default gateway		STRING(32)
5	DNS server		STRING(32)
6	DNS name		STRING(50)
7	Rx packets		UNSIGNED_32
8	Tx packets		UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

3.6.4 GCI-SDO communication (TCP port 9410)

The TCP port 9410 serves to establish a parameter communication with the i700 servo inverter. The PC (client) used for parameter setting and the i700 servo inverter (server) communication with each other by exchanging data via the Ethernet telegrams which are fed into the cyclic EtherCAT telegrams (see also System architecture).

- The parameter data of the i700 servo inverter are stored in "objects". The objects serves, for instance, to set operating parameters and motor data or query diagnostic information.
- The parameter data are transmitted as SDOs (Service Data Objects) and confirmed by the receiver, i.e. the transmitter receives a feedback whether the transmission was successful.
- The transmission of the parameter data usually is not time-critical.
- The parameter communication enables the writing and reading access to the object directory of the i700 servo inverter.

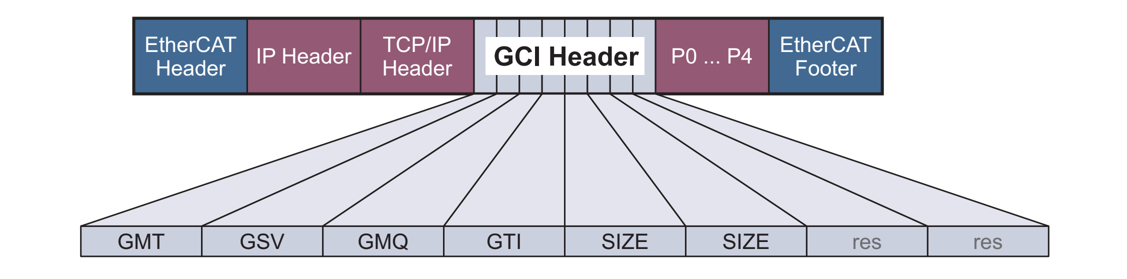
Note!

- Only one communication connection at a time is possible via the TCP port 9410, i.e. only one client at a time can be connected to the i700 servo inverter.
- In case of a writing access to parameter data, make sure that the changes made are not automatically saved in the i700 servo inverter.
  The »EASY Starter« serves to upload the parameters of the i700 servo inverter and save them as a file. This file can then be imported to the Engineering tool (e.g. »PLC Designer«).

3.6.4.1 Structure of the EtherCAT data telegram

The GCI protocol is used for communication.

The EtherCAT data telegram is shown below. Here, the GCI header represents the part of the program that is independent of the type of command transmitted.

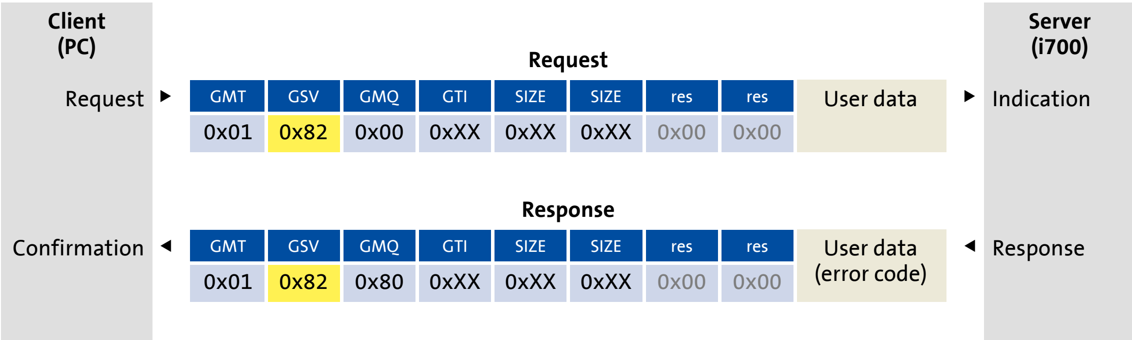


[3-2] Structure of the GCI header within the EtherCAT frame

Field	Size	Description	
GMT	1 byte	GCI message type	
		0x01	Reserved
GSV	1 byte	GCI service identification	
		0x82	Read parameters
		0x83	Write parameters
GMQ	1 byte	GCI message qualifier	
		Bit 7	rsp (Request/Response)
		0	Request
		1	Response
		Bit 6	a (Abort)
		0	Data transfer OK
		1	Abort of the data transfer • The transfer is either aborted by the client or the server of a parameter data telegram. • A message is aborted without any confirmation. If the client waits for its message to be confirmed, it will receive the abort notice instead.
		Bit 5 ... bit 0	res (reserved)
		0b000000	Data contents = 0
GTI	1 byte	GCI transaction ID	
		0x00 ... 0xFF	Serial number (transaction identification) • For each client a definite serial number (0 ... 255) is allocated. • The serial number in the multitasking environment is used for referencing to the calling tasks (reverse transaction).
SIZE	2 bytes	Length of the user data • The user data area or the data telegram contains the parameter data. ▸ Assignment of user data areas P0 ... P4	
		0x14	20 bytes
		...	...
		0x114	276 bytes
res	2 bytes	Reserved	
		0x0000	Data contents = 0

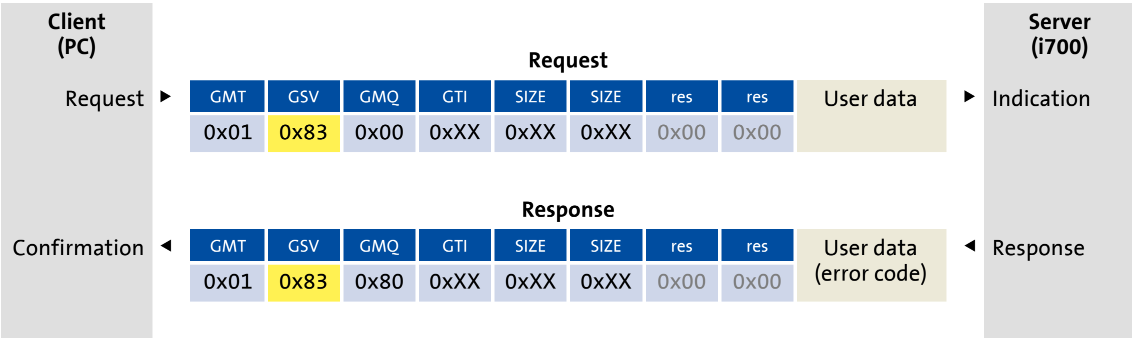
Read parameters

The service identification (GSV) = 0x82 in the GCI header serves to read parameter data from the i700 servo inverter:



Write parameters

The service identification (GSV) = 0x83 in the GCI header serves to write parameter data to the i700 servo inverter:



3.6.4.2 Assignment of user data areas P0 ... P4

Range	Byte 1	Byte 2	Byte 3	Byte 4
P0	Status/error code		Data type	Reserved
P1	Index		Reserved	Reserved
P2	Subindex		Reserved	Reserved*
P3	Parameter value			
P4	Parameter value			

* When the data type VISIBLE_STRING is transmitted, byte 4 contains the number of the characters attached.

Data type in P0 / byte 3

ID	Data type	Data length
0x01	INTEGER_8	1 byte
0x02	INTEGER_16	2 bytes
0x03	INTEGER_32	4 bytes
0x04	INTEGER_64	8 bytes
0x05	UNSIGNED_8	1 byte
0x06	UNSIGNED_16	2 bytes
0x07	UNSIGNED_32	4 bytes
0x08	UNSIGNED_64	8 bytes
0x09	FLOATING_POINT	4 bytes
0x0A	VISIBLE_STRING	max. 256 bytes
0x0B	OCTET_STRING	max. 256 bytes

Parameter value in P3 and P4

Depending on the data type, the parameter value assigns 1 ... 8 bytes. The data is stored in the Little-Endian format, i.e. first the low byte or low word, then the high byte or high word:

Data length of parameter value	Data area P3				Data area P4			
	Byte 1	Byte 2	Byte 3	Byte 4	Byte 1	Byte 2	Byte 3	Byte 4
1 byte	**Value**	0x00	0x00	0x00	0x00	0x00	0x00	0x00
2 bytes	Low byte	High byte	0x00	0x00	0x00	0x00	0x00	0x00
	Value							
4 bytes	Double word				0x00	0x00	0x00	0x00
	Low word		High word					
	Low byte	High byte	Low byte	High byte				
	Value							
8 bytes	Lower-order double word				Higher-order double word			
	Low word		High word		Low word		High word	
	Low byte	High byte	Low byte	High byte	Low byte	High byte	Low byte	High byte
	Value							

3.6.4.3 Error codes

The error code is located in the User data area P0, byte 1 and byte 2.

User data area P0			
Byte 1	**Byte 2**	**Byte 3**	**Byte 4**
Error code		Data type	Reserved
Example error code 0x9002			
Low byte	High byte		
0x02	0x90		

Note!

The other user data contents correspond to those of an error-free message.

Possible error codes

Error code		Definition	Description
Decimal	hex		
33803	0x840B	Invalid type	Invalid parameter type
33804	0x840C	Limit violation	Invalid parameter value
33806	0x840E	Unknown parameter	Invalid parameter index
33812	0x8414	Invalid size	Invalid parameter format
33813	0x8415	Not in select list	Parameter is not in the selection list
33814	0x8416	Read not allowed	Parameter read is not allowed
33815	0x8417	Write not allowed	Parameter write is not allowed
33816	0x8418	CINH not set	Controller inhibit is not set
33829	0x8425	Invalid subindex	Invalid parameter subindex
33837	0x842D	Access not allowed	Parameter access not allowed
36873	0x9009	Wrong GMT received	The general telegram identification does not correspond to the GCI communication.
36874	0x900A	Unknown server request	Internal error in the GCI
36878	0x900E	SRV timeout	

3.6.4.4 Telegram example 1: Querying the heatsink temperature (read request)

The heatsink temperature of the i700 servo inverter is to be read.

- Object to be read: 0x2D84:1
- Assumption: ϑ = 43°C

Request

- SDO command (GSV) = 0x82 = "Read parameter"
- GCI Message Qualifier (GMQ) = 0x00 = 0b00000000 = "Request"
- Transaction ID (GTI) here "0" (optional consecutive number 0 ... 255)
- Length of the user data (SIZE) = 0x0014 = 20 bytes

GCI header							
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
GMT	GSV	GMQ	GTI	SIZE	SIZE	res	res
0x01	0x82	0x00	0x00	0x14	0x00	0x00	0x00
Fixed	Read parameters	Request	Transactions ID	Length of the user data = 20 bytes		Reserved	

User data area P0			
Byte 1	Byte 2	Byte 3	Byte 4
Reserved		Data type	Reserved
0x00	0x00	0x00	0x00
		Optional for read request	

User data area P1				User data area P2			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 1	Byte 2	Byte 3	Byte 4
Index		Reserved		Subindex		Reserved	
0x84	0x2D	0x00	0x00	0x01	0x00	0x00	0x00
Index = 0x2D84				Subindex = 1			

User data area P3				User data area P4			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 1	Byte 2	Byte 3	Byte 4
Reserved				Reserved			
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00

Response

GCI Message Qualifier (GMQ) = 0x80 = 0b10000000 = "Response"

GCI header							
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
GMT	GSV	GMQ	GTI	SIZE	SIZE	res	res
0x01	0x82	0x80	0x00	0x14	0x00	0x00	0x00
Fixed	Read parameters	Response	Transactions ID	Length of the user data = 20 bytes		Reserved	

User data area P0			
Byte 1	Byte 2	Byte 3	Byte 4
Reserved		Data type	Reserved
0x00	0x00	0x02	0x00
		INTEGER_16	

User data area P1				User data area P2			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 1	Byte 2	Byte 3	Byte 4
Index		Reserved		Subindex		Reserved	
0x84	0x2D	0x00	0x00	0x01	0x00	0x00	0x00
Index = 0x2D84				Subindex = 1			

User data area P3				User data area P4			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 1	Byte 2	Byte 3	Byte 4
Parameter value		Reserved					
0xAE	0x01	0x00	0x00	0x00	0x00	0x00	0x00
Read value = 0x01AE = 430 = 43.0 [°C]							

3.6.4.5 Telegram example 2: Querying the software version of the i700 (read request)

The software version of the i700 servo inverter is to be read.

- Object to be read: 0x100A
- Assumption: Software version = "1.5.0.9999 (release)"

Request

- SDO command (GSV) = 0x82 = "Read parameter"
- GCI Message Qualifier (GMQ) = 0x00 = 0b00000000 = "Request"
- Transaction ID (GTI) here "1" (optional consecutive number 0 ... 255)
- Length of the user data (SIZE) = 0x0014 = 20 bytes

GCI header							
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
GMT	GSV	GMQ	GTI	SIZE	SIZE	res	res
0x01	0x82	0x00	0x01	0x14	0x00	0x00	0x00
Fixed	Read parameters	Request	Transactions ID	Length of the user data = 20 bytes		Reserved	

User data area P0			
Byte 1	Byte 2	Byte 3	Byte 4
Reserved		Data type	Reserved
0x00	0x00	0x00	0x00
		Optional for read request	

User data area P1				User data area P2			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 1	Byte 2	Byte 3	Byte 4
Index		Reserved		Subindex		Reserved	
0x0A	0x10	0x00	0x00	0x00	0x00	0x00	0x00
Index = 0x100A				No subindex			

User data area P3				User data area P4			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 1	Byte 2	Byte 3	Byte 4
Reserved				Reserved			
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00

Response

- GCI Message Qualifier (GMQ) = 0x80 = 0b10000000 = "Response"
- Length of the user data =
  20 bytes of standard user data area P0 ... P4 **plus**
  21 bytes of attached string (incl. \0 termination) = 41 bytes

GCI header							
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
GMT	GSV	GMQ	GTI	SIZE	SIZE	res	res
0x01	0x82	0x80	0x01	0x29	0x00	0x00	0x00
Fixed	Read parameters	Response	Transactions ID	Length of the user data = 41 bytes		Reserved	

User data area P0			
Byte 1	Byte 2	Byte 3	Byte 4
Reserved		Data type	Reserved
0x00	0x00	0x0A	0x00
		VISIBLE_STRING	

User data area P1				User data area P2			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 1	Byte 2	Byte 3	Byte 4
Index		Reserved		Subindex		Reserved	
0x0A	0x10	0x00	0x00	0x00	0x00	0x00	0x00
Index = 0x100A				No subindex			

User data area P3				User data area P4			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 1	Byte 2	Byte 3	Byte 4
Reserved				Reserved			
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00

Note!

The read parameter value of data type VISIBLE_STRING (here: "1.5.0.9999 (release)") follows subsequent to the standard user data area.

3.6.4.6 Telegram example 3: Setting the LV warning threshold in the i700 (write request)

The warning threshold for the low-voltage detection (LV) is to be set in the i700 servo inverter to 400 V.

- Object to be written: 0x2540:2

Request

- SDO command (GSV) = 0x83 = "Write parameter"
- GCI Message Qualifier (GMQ) = 0x00 = 0b00000000 = "Request"
- Transaction ID (GTI) here "42" (optional consecutive number 0 ... 255)
- Length of the user data (SIZE) = 0x0014 = 20 bytes

GCI header							
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
GMT	GSV	GMQ	GTI	SIZE	SIZE	res	res
0x01	0x83	0x00	0x2A	0x14	0x00	0x00	0x00
Fixed	Write parameters	Request	Transactions ID	Length of the user data = 20 bytes		Reserved	

User data area P0			
Byte 1	Byte 2	Byte 3	Byte 4
Reserved		Data type	Reserved
0x00	0x00	0x06	0x00
		UNSIGNED_16	

User data area P1				User data area P2			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 1	Byte 2	Byte 3	Byte 4
Index		Reserved		Subindex		Reserved	
0x40	0x25	0x00	0x00	0x02	0x00	0x00	0x00
Index = 0x2540				Subindex = 2			

User data area P3				User data area P4			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 1	Byte 2	Byte 3	Byte 4
Parameter value		Reserved					
0x90	0x01	0x00	0x00	0x00	0x00	0x00	0x00
Value to be written = 0x0190 = 400 [V]							

Response

GCI Message Qualifier (GMQ) = 0x80 = 0b10000000 = "Response"

GCI header							
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
GMT	GSV	GMQ	GTI	SIZE	SIZE	res	res
0x01	0x83	0x80	0x2A	0x14	0x00	0x00	0x00
Fixed	Write parameters	Response	Transactions ID	Length of the user data = 20 bytes		Reserved	

User data area P0			
Byte 1	Byte 2	Byte 3	Byte 4
Reserved		Data type	Reserved
0x00	0x00	0x06	0x00
		UNSIGNED_16	

User data area P1				User data area P2			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 1	Byte 2	Byte 3	Byte 4
Index		Reserved		Subindex		Reserved	
0x40	0x25	0x00	0x00	0x02	0x00	0x00	0x00
Index = 0x2540				Subindex = 2			

User data area P3				User data area P4			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 1	Byte 2	Byte 3	Byte 4
Parameter value		Reserved					
0x90	0x01	0x00	0x00	0x00	0x00	0x00	0x00
Written value (reflected)							

4 Device settings

Objects described in this chapter

Object	Name	Data type
0x10F1	ECAT: Behaviour in case of error	RECORD
0x2000	Device: Data	RECORD
0x2001	Device: Name	STRING(128)
0x2021	Device: Optical recognition	RECORD
0x2022	Device command	UNSIGNED_32
0x2540	Device: Voltage values	RECORD
0x2580	ECAT DC: Real-time information	RECORD

4.1 Behaviour in case of error

The i700 servo inverter features three different EtherCAT monitoring modes:

- Sync0 monitoring when DC mode is used
- PDO frame failure detection when DC mode is used
- Monitoring to EtherCAT line interruption

Sync0 monitoring when DC mode is used

This monitoring mode checks whether the Sync0 Signals are generated at the correct time in the i700 servo inverter if the "Distributed Clock mode" (DC mode) has been selected and the i700 servo inverter is in the "Operational" status.

- If no Sync0 signals arrive anymore during double the Sync0 cycle time, the i700 servo inverter changes to the "Safe-Operational" status and triggers an error (CiA402 error code 0x8700). 0x32 is returned as bus status (AL status code).
- After "Pre-Operational" has changed to "Safe-Operational", the generation of Sync0 pulses has to be started within 5 seconds. If this is not the case, or if a change from "Safe-Operational" to "Operational" is requested without the signals being generated accordingly, this error is triggered as well.
- This monitoring mode cannot be configured.

PDO frame failure detection when DC mode is used

This monitoring mode checks whether an EtherCAT-PDO telegram (Sync Manager 2 Event) has arrived between two Sync0 signals if the "Distributed Clock mode" (DC mode) has been selected. For this purpose, the i700 servo inverter is provided with an internal frame failure error counter which is increased by the value "3" in case of a frame failure. For every PDO received correctly, the error counter is reduced by the value "1".

This monitoring mode can be configured via the 0x10F1:2 object:

- From version 01.04, monitoring is activated in the Lenze setting (0x10F1:2 = "20").
- If a value higher than "0" is set in 0x10F1:2: If the internal frame failure error counter reaches the set value, the i700 servo inverter changes to the "Safe-Operational" status and triggers an error (CiA402 error code 0x8700).

Tip!

It is reasonable to set a value ≥"4" in 0x10F1:2 to tolerate a failed PDO and prevent two PDO failures in a row.

The following table lists some possible settings:

Permitted PDO failures in a row	Monitoring threshold (0x10F1:2)
0	1 ... 2
1	4 ... 5
2	7 ... 8
3	10 ... 11

From version 01.02, an extrapolator is available in the i700 servo inverter for the motor control which can extrapolate the setpoints for one cycle. If two PDO frames fail in a row, no extrapolation takes place anymore. The last value is frozen.

Monitoring to EtherCAT line interruption

This monitoring mode generally checks whether there is an EtherCAT line interruption.

- If this is the case, the i700 servo inverter changes to the "Safe-Operational" status and triggers an error (CiA402 error code 0x8181).
- This monitoring mode cannot be configured and functions with and without "Distributed Clock mode" (DC mode).

0x10F1 - ECAT: Behaviour in case of error

Response of the device in the event of an error

Sub.	Name	Lenze setting	Data type
▸ 1	Local Error Reaction	2: Device-specific status	UNSIGNED_32
▸ 2	Synchronisation: Error threshold	20	UNSIGNED_32

Subindex 1: **Internal device response**			
Selection list (Lenze setting printed in bold)		Info	
2	**Device-specific status**	An error response is carried out by the i700 servo inverter only.	
☑ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			UNSIGNED_32

Subindex 2: **Synchronisation: Error threshold**				
Setting for PDO frame failure detection • If the internal frame failure error counter reaches the value set here, the Servo-Inverter i700 changes to the "Safe-Operational" status and triggers an error (CiA402 error code 0x8700). • From version 01.04, the PDO frame failure detection is activated in the Lenze setting "20".				
Setting range (min. value | unit | max. value)			Lenze setting	
0		32	20	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_32

4.2 Device identification data

0x2000 - Device: Data

Type code (Lenze ID), serial number and manufacturing date of the device

Sub.	Name	Lenze setting	Data type
1	Device: Product designation		STRING(50)
2	Device: Serial number		STRING(50)
3	Device: Manufacturing date		STRING(50)
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x2001 - Device: Name

Any device name (e.g. "Wheel drive") can be set in this object for the purpose of device identification.

☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX		STRING(128)

4.3 Function "Optical device recognition"

0x2021 - Device: Optical recognition

In the case of applications with multiple interconnected controllers, it may be difficult to locate a device that has been connected online. The "Optical device recognition" function serves to locate the i700 servo inverter by means of blinking LEDs.

- A setting of "1: Start" in subindex 1 activates the function:
  - The two LEDs "RDY" and "ERR" on the front of the i700 servo inverter will blink for the time period set in subindex 2 with a blinking frequency of 20 Hz. Then the function is deactivated automatically.
  - The LEDs on the RJ45 sockets are not used for this function.
  - If the function is reactivated within the time set, the time is extended correspondingly.
- With the setting "0: Stop" in subindex 1 the function can be aborted/deactivated prematurely.

Sub.	Name	Lenze setting	Data type
▸ 1	Start optical recognition	0: Stop	UNSIGNED_8
▸ 2	Optical recognition: Blinking time	5 s	UNSIGNED_16

Subindex 1: **Start optical recognition**			
Selection list (Lenze setting printed in bold)			
0	**Stop**		
1	Start		
☑ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			UNSIGNED_8

Subindex 2: **Optical recognition: Blinking time**				
Setting range (min. value | unit | max. value)			Lenze setting	
0	s	6000	5 s	
☑ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX				UNSIGNED_16

4.4 Device commands

Note!

The execution of a device command may lead to an interruption of the EtherCAT communication with the master and to a standstill of the axis!

0x2022 - Device command

Selection list (Lenze setting printed in bold)		Info	
0	**No command**		
1000	Load Lenze setting	Load default parameter set	
1100	Restart device	Device restart	
☑ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			UNSIGNED_32

4.5 Monitoring of the DC-bus voltage

For the device as a whole, the DC-bus voltage is monitored with regard to undervoltage and overvoltage.

- The voltage monitoring is active in the device statuses "Ready to switch on", "Switched on", "Operation enabled" and "Quick stop is active".
- The warning thresholds for the monitoring are adjustable and preset for a rated mains voltage of 400 V in the Lenze setting.
- The error thresholds for the monitoring result from the set rated mains voltage:

Mains	Undervoltage thresholds			Overvoltage thresholds		
Rated voltage 0x2540:1	Warning threshold 0x2540:2	Error threshold 0x2540:3	Threshold "Reset error" 0x2540:4	Warning threshold 0x2540:5	Error threshold 0x2540:6	Threshold "Reset error" 0x2540:7
60 V	430 V	10 V	15 V	795 V	60 V	50 V
230 V_{eff}		200 V	225 V		800 V	790 V
400 V_{eff}		285 V	430 V		800 V	790 V
480 V_{eff}		490 V	535 V		800 V	790 V
Greyed out = read access only or threshold specified						

0x2540 - Device: Voltage values

Sub.	Name	Lenze setting	Data type
▸ 1	Mains: Rated voltage	1: 400 Veff	UNSIGNED_8
▸ 2	Undervoltage (LU): Warning threshold	430 V	UNSIGNED_16
▸ 3	Undervoltage (LU): Error threshold	285 V	UNSIGNED_16
▸ 4	Undervoltage (LU): Threshold 'Reset error'	430 V	UNSIGNED_16
▸ 5	Overvoltage (OU): Warning threshold	795 V	UNSIGNED_16
▸ 6	Overvoltage (OU): Error threshold	800 V	UNSIGNED_16
▸ 7	Overvoltage (OU): Threshold 'Reset error'	790 V	UNSIGNED_16

Subindex 1: **Mains: Rated voltage**		
Selection of the mains voltage used with which the i700 servo inverter is operated.		
Selection list (Lenze setting printed in bold)		
0	230 Veff	
1	**400 Veff**	
2	480 Veff	
3	Reserved	
4	60 V (setting-up operation)	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX		UNSIGNED_8

Subindex 2: **Undervoltage (LU): Warning threshold**			
Warning threshold for monitoring with regard to undervoltage • If the DC-bus voltage of the i700 servo inverter falls below the threshold value set here, the device reports a warning. Reset is effected with a hysteresis of 10 V.			
Setting range (min. value | unit | max. value)			Lenze setting
0	V	800	430 V
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_16

Subindex 3: **Undervoltage (LU): Error threshold**				
Display of the error threshold for monitoring with regard to undervoltage • If the DC-bus voltage of the i700 servo inverter falls below the threshold value shown here, the device reports an error and the motor module changes to the "Fault" error status. An automatic restart after mains recovery is not possible.				
Display area (min. value | unit | max. value)			Initialisation	
0	V	800	285 V	
☐ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

Subindex 4: **Undervoltage (LU): Threshold 'Reset error'**				
Display of the error reset threshold for monitoring with regard to undervoltage				
Display area (min. value | unit | max. value)			Initialisation	
0	V	800	430 V	
☐ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

Subindex 5: **Overvoltage (OU): Warning threshold**				
Warning threshold for monitoring with regard to overvoltage • If the DC-bus voltage of the i700 servo inverter exceeds the threshold value set here, the device reports a warning. Reset is effected with a hysteresis of 10 V.				
Setting range (min. value | unit | max. value)			Lenze setting	
0	V	800	795 V	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

Subindex 6: **Overvoltage (OU): Error threshold**				
Display of the error threshold for monitoring with regard to overvoltage • If the DC-bus voltage of the i700 servo inverter exceeds the threshold value shown here, the device reports an error and the motor module changes to the "Fault" error status.				
Display area (min. value | unit | max. value)			Initialisation	
0	V	800	800 V	
☐ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

Subindex 7: **Overvoltage (OU): Threshold 'Reset error'**				
Display of the error reset threshold for monitoring with regard to overvoltage				
Display area (min. value | unit | max. value)			Initialisation	
0	V	800	790 V	
☐ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

Related topics:

▸ 24-V supply voltage monitoring (📖 235)

4.6 Real-time information (distributed clock)

0x2580 - ECAT DC: Real-time information

All time information provided in this object is based on UTC and transmitted in the format defined by EtherCAT for this purpose: In nanoseconds, at a width of 64 bits, based on a date of January 01, 2000, (2000-01-01) and a time of 00:00.
There are various ways for the i700 servo inverter to receive its real-time information from the outside:
- From the (EtherCAT) master (DC synchronous in the EtherCAT state "Safe-Operational" or "Operational").
- By writing to subindex 4 (ECAT DC: Current time).

Sub.	Name	Lenze setting	Data type
1	ECAT DC: Status real-time information	0: No real-time information received yet	UNSIGNED_8
2	ECAT DC: Time stamp first real-time information		UNSIGNED_64
3	ECAT DC: Time stamp last real-time information		UNSIGNED_64
4	ECAT DC: Current time		UNSIGNED_64

Subindex 1: **ECAT DC: Status real-time information**		
Selection list (read only)		Info
0	**No real-time information received yet**	The i700 servo inverter has not yet received any real-time information from the outside since switch-on. • The i700 servo inverter is still operating with a time which is based on the time when the device was switched off last or which is based on the time stamp of the firmware (whichever time is later).
1	Receive at least one piece of real-time information	The i700 servo inverter has at least once received one piece of real-time information from the outside since switch-on. • The first and the last time the i700 servo inverter has received real-time information are displayed in subindices 2 and 3.
2	Synchronous - cyclically updated	The i700 servo inverter receives cyclically updated real-time information from the outside. • The i700 servo inverter reports this status if it is e.g. DC synchronous in the EtherCAT state "Safe-Operational" or "Operational".
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX		UNSIGNED_8

Subindex 2: **ECAT DC: Time stamp first real-time information**			
Display of the time when the i700 servo inverter received its first real-time information from the outside since switch-on. • In the "No real-time information received yet" status, a value of "0" is displayed.			
Display area (min. value | unit | max. value)			Initialisation
0	ns	2^{64}-1	
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			UNSIGNED_64

Subindex 3: **ECAT DC: Time stamp last real-time information**			
Display of the time when the i700 servo inverter received its last real-time information from the outside.			
Display area (min. value | unit | max. value)			Initialisation
0	ns	2^{64}-1	
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			UNSIGNED_64

<table>
<tr><td colspan="4">Subindex 4: ECAT DC: Current time</td></tr>
<tr><td colspan="4">Display of the time information the i700 servo inverter is currently using (time of the device if you will).
• The contents are updated every time this subindex is read.
• Depending on the status (see subindex 1), this time information is more or less precise:
  • In the "No real-time information received yet" status, the clock is generally slow compared to real time. This may mean weeks or even months if the i700 servo inverter was switched off for a long time.
  • In the "Receive at least one piece of real-time information" status, the clock is accurate to the second unless the time of the last time information (subindex 3) does not date back several days.
  • In the "Synchronous - cyclically updated" status, the clock is accurate to the µs with the clock of the higher-level master.
  • Due to the synchronisation with an external master, this clock is also able to go backwards. TwinCAT uses the PC clock e.g. as its time base!
Setting the time:
• The clock is set by writing a value.
  • If the i700 servo inverter is in the "No real-time information received yet" status before the writing process, it will then change to the "Receive at least one piece of real-time information" status.
  • If the i700 servo inverter is in the "Synchronous - cyclically updated" status during the writing process, it will ignore the written value.</td></tr>
<tr><td colspan="3">Setting range (min. value | unit | max. value)</td><td>Lenze setting</td></tr>
<tr><td>0</td><td>ns</td><td>2^{64}-1</td><td></td></tr>
<tr><td colspan="3">☑ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX</td><td>UNSIGNED_64</td></tr>
</table>

5 Motor control & motor settings

This chapter describes the commissioning of the motor control in a recommended sequence.

Objects described in this chapter

Object Axis A	Object Axis B	Name	Data type
Axis control			
0x2822	0x3022	Axis command	UNSIGNED_32
0x2823	0x3023	Axis command: Progress	UNSIGNED_8
0x2825	0x3025	Modes of operation	UNSIGNED_8
0x2832	0x3032	Identification: Status word	UNSIGNED_16
0x2835	0x3035	Manual test mode: Settings	RECORD
0x2836	0x3036	Manual jog: Settings	RECORD
Controller settings			
0x2900	0x3100	Speed controller: Parameter	RECORD
0x2901	0x3101	Speed controller: Gain - adaption	UNSIGNED_16
0x2902	0x3102	Speed controller: Load value	INTEGER_16
0x2903	0x3103	Speed: Speed setpoint - filter time	UNSIGNED_16
0x2904	0x3104	Speed: Actual speed - filter time	UNSIGNED_16
0x2910	0x3110	Moments of inertia	RECORD
0x2939	0x3139	Switching frequency	UNSIGNED_8
0x2941	0x3141	Current controller: Feedforward control	UNSIGNED_8
0x2942	0x3142	Current controller: Parameter	RECORD
0x2943	0x3143	Motor: Current setpoint - filter time	UNSIGNED_16
0x2944	0x3144	Torque: Notch filter torque setpoint	RECORD
0x2945	0x3145	Torque: Jerk limitation setpoint	UNSIGNED_16
0x2947	0x3147	Inverter characteristic: Voltage grid points (y)	RECORD
0x2980	0x3180	Position controller: Gain	UNSIGNED_32
0x2981	0x3181	Position controller: Gain - adaption	UNSIGNED_16
0x2982	0x3182	Position controller: Output signal limitation	UNSIGNED_32
0x2983	0x3183	Position: Select new actual position	INTEGER_32
0x2984	0x3184	Determine target position: Mode	UNSIGNED_8
0x29C0	0x31C0	Field controller: Parameter	RECORD
0x29E0	0x31E0	Field weakening controller: Parameter	RECORD
0x29E1	0x31E1	Field set value limitation	UNSIGNED_16
0x29E2	0x31E2	DC link circuit voltage: Filter time	UNSIGNED_16
0x29E3	0x31E3	Motor voltage act. value: Filter time	UNSIGNED_16
0x29E4	0x31E4	Voltage reserve range	UNSIGNED_8
V/f operation			
0x2B00	0x3300	VFC: V/f characteristic - shape	UNSIGNED_8
0x2B01	0x3301	VFC: V/f characteristic - define reference point	RECORD
0x2B02	0x3302	VFC: User-definable V/f characteristic - frequency grid points (x)	RECORD
0x2B03	0x3303	VFC: User-definable V/f characteristic - voltage grid points (y)	RECORD
0x2B04	0x3304	VFC: Voltage vector control - current setpoint	UNSIGNED_32

Object		Name	Data type
Axis A	Axis B		
0x2B05	0x3305	VFC: Voltage vector control parameter	RECORD
0x2B06	0x3306	VFC: Voltage boost	UNSIGNED_16
0x2B07	0x3307	VFC: Load adjustment - parameter	RECORD
0x2B08	0x3308	VFC: Imax controller - parameter	RECORD
0x2B09	0x3309	VFC: Slip compensation - parameter	RECORD
0x2B0A	0x330A	VFC: Oscillation damping - parameter	RECORD
0x2B0B	0x330B	VFC: Frequency setpoint	INTEGER_16
0x2B0C	0x330C	VFC: Override point of field weakening	INTEGER_16
0x2B80	0x3380	DC-injection braking: Current	UNSIGNED_16
0x2BA0	0x33A0	Flying restart: Activate	UNSIGNED_8
0x2BA1	0x33A1	Flying restart: Current	UNSIGNED_16
0x2BA2	0x33A2	Flying restart: Start frequency	INTEGER_16
0x2BA3	0x33A3	Flying restart: Integration time	UNSIGNED_16
0x2BA4	0x33A4	Flying restart: Min. deviation	UNSIGNED_16
0x2BA5	0x33A5	Flying restart: Delay time	UNSIGNED_16
0x2BA6	0x33A6	Flying restart: Result	RECORD
Motor settings			
0x2C00	0x3400	Motor control	UNSIGNED_8
0x2C01	0x3401	Motor: Common parameters	RECORD
0x2C02	0x3402	Motor (ASM): Parameter	RECORD
0x2C03	0x3403	Motor (SM): Parameter	RECORD
0x2C04	0x3404	Motor: Lss saturation characteristic - inductance grid points (y)	RECORD
0x2C05	0x3405	Motor: Lss saturation charcteristic - reference for current grid points (x)	UNSIGNED_16
0x2C06	0x3406	Motor (SM): Magnet characteristic (current) - grid points	RECORD
0x2C07	0x3407	Motor (ASM): Lh saturation characteristic - inductance grid points (y)	RECORD
0x2C08	0x3408	Motor: Motor parameter setting method	UNSIGNED_8
0x6075	0x6875	Motor rated current	UNSIGNED_32
0x6076	0x6876	Motor rated torque	UNSIGNED_32
Feedback system			
0x2C40	0x3440	Encoder: Type	UNSIGNED_8
0x2C41	0x3441	Hiperface: Parameter	RECORD
0x2C42	0x3442	Encoder: Parameter	RECORD
0x2C43	0x3443	Resolver: Number of pole pairs	UNSIGNED_8
0x2C44	0x3444	Resolver error compensation: Parameter	RECORD
0x2C45	0x3445	Open circuit in feedback system: Response	UNSIGNED_8
0x2C46	0x3446	Feedback system: Specifiable number of revolutions	UNSIGNED_16
0x2C5F	0x345F	Feedback system: Parameter CRC	UNSIGNED_32
Pole position identification			
0x2C60	0x3460	Monitoring pole position identification: Response	UNSIGNED_8
0x2C61	0x3461	Pole position identification PPI (360°)	RECORD
0x2C62	0x3462	Pole position identification PPI (min. movement)	RECORD
0x2C63	0x3463	Pole position identification PPI (without movement)	RECORD
Advanced settings			
0x2DE0	0x35E0	Advanced settings	RECORD

5.1 Required commissioning steps (short overview)

The following subchapters provide information on the individual commissioning steps required for a specific control mode/motor type combination.

5.1.1 Servo control for synchronous motor (SM)

Required commissioning steps	
1.	Wiring check by means of manual test modes
2.	Setting the control mode
3.	Accepting/adapting plant parameters
4.	Setting the motor parameters for the servo control
5.	Set motor monitoring: • Monitoring of the motor utilisation (I^2xt) • Motor temperature monitoring
6.	Setting the feedback system for the servo control
7.	Only required for other manufacturers' motors: • Setting and optimising the current controller • Correction of the stator leakage inductance (Lss)... • Synchronous motor (SM): Pole position identification
8.	Only required for the automatic calculation of the speed controller parameters: Determining the total moment of inertia
9.	Setting the speed controller
10.	Setting the position controller

Optional commissioning steps (additional function or fine adjustment)	
A.	Compensating for inverter influence on output voltage
B.	Synchronous motor (SM): Compensating for temperature and current influences
C.	Jerk limitation
D.	Notch filters (band-stop filters)
E.	"Short-circuit braking" function

5.1.2 Servo control for asynchronous motor (ASM)

Required commissioning steps	
1.	Wiring check by means of manual test modes
2.	Setting the control mode
3.	Accepting/adapting plant parameters
4.	Setting the motor parameters for the servo control
5.	Set motor monitoring: • Monitoring of the motor utilisation (I²xt) • Motor temperature monitoring
6.	Setting the feedback system for the servo control
7.	Only required for other manufacturers' motors: Setting and optimising the current controller
8.	Only required for the automatic calculation of the speed controller parameters: Determining the total moment of inertia
9.	Setting the speed controller
10.	Setting the position controller
11.	Only required for other manufacturers' motors: • Setting the field controller (ASM) • Setting the field weakening controller (ASM)

Optional commissioning steps (additional function or fine adjustment)	
A.	Compensating for inverter influence on output voltage
B.	Correction of the stator leakage inductance (Lss)...
C.	Asynchronous motor (ASM): Identifying the Lh saturation characteristic
D.	Estimating the optimal magnetising current
E.	Jerk limitation
F.	Notch filters (band-stop filters)
G.	"DC-injection braking" function

5.1.3 V/f characteristic control for asynchronous motor (ASM)

Required commissioning steps	
1.	Wiring check by means of manual test modes
2.	Setting the control mode
3.	Accepting/adapting plant parameters
4.	Compensating for inverter influence on output voltage
5.	Only required if voltage vector control, DC-injection braking, or flying restart process is activated: Setting and optimising the current controller
6.	Defining the V/f characteristic shape
7.	Setting the voltage boost
8.	Setting the load adjustment
9.	Activating the voltage vector control (Imin controller)
10.	Defining the behaviour at the current limit (Imax controller)

Optional commissioning steps (additional function or fine adjustment)	
A.	"Flying restart" function
B.	"DC-injection braking" function
C.	Setting the slip compensation
D.	Setting the oscillation damping

5.2 Commissioning functions (short overview)

For a quick commissioning, the i700 servo inverter provides various functions which serve to automatically calculate and set the parameters. These functions can be executed via the following object:

0x2822 | 0x3022 - Axis command

Via the cross-reference in the information column, a detailed description of the respective function is provided.

Selection list (Lenze setting printed in bold)		Info	
0	**No command**		
1	Inverter characteristic: Load Lenze setting	▶ Loading the standard inverter characteristic	
2	Estimating the optimal magnetising current	▶ Estimating the optimal magnetising current	
3	Motor model parameters: Determine from motor data by approximation	▶ Enter nameplate data and have motor model parameters determined automatically	
4	Current controller: Calculate controller parameters	▶ Calculate current controller parameters	
5	Speed controller: Calculate controller parameters	▶ Calculate speed controller parameters	
6	Position controller: Calculate controller parameters	▶ Calculate position controller parameters	
7	Field controller: Calculate controller parameters	▶ Calculate field controller parameters	
8	Field weakening controller: Calculate controller parameters	▶ Calculate field weakening controller parameters	
9	Identify resolver error	▶ Identify resolver error	
10	VFC: Calculate Imin controller parameters	▶ V/f operation: Activating the voltage vector control	
11	VFC: Calculate Imax controller parameters	▶ V/f operation: Defining the behaviour at the current limit (Imax controller)	
12	VFC: Calculate flying restart controller parameters	▶ V/f operation: "Flying restart" function	
13	Lh saturation characteristic : Load Lenze setting	▶ Load standard Lh saturation characteristic	
14	Get Hiperface information from encoder	From version 01.03 Read type code, number of increments and number of distinguishable revolutions out of the encoder and automatically enter them into the corresponding Hiperface parameters. ▶ Additional settings for SinCos absolute value encoder with HIPERFACE® protocol	
☑ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			UNSIGNED_32

0x2823 | 0x3023 - Axis command: Progress

Display of the current progress of the activated commissioning function

Display area (min. value | unit | max. value)			Initialisation	
0		100	0	
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX				UNSIGNED_8

0x2825 | 0x3025 - Axis: Operating mode

This object serves to activate different commissioning test modes and procedures for the automatic parameter identification.
Via the cross-reference in the information column, a detailed description of the respective function is provided.

Selection list (Lenze setting printed in bold)		Info	
0	**CiA402 mode active**	Normal operation (CiA402 operating modes active) ▶ CiA402 device profile	
1	Manual test mode: Voltage/frequency	▶ Test mode "Voltage/frequency"	
2	Manual test mode: Current/frequency	▶ Test mode "Current/frequency"	
3	Manual test mode: Current pulse	▶ Test mode "Current pulse"	
4	Manual control mode	▶ Manual control	
5	Pole position identification PPI (360°)	▶ Synchronous motor (SM): Pole position identification • PLI without motion from version 01.03.	
6	Pole position identification PPI (min. movement)		
7	Pole position identification PPI (without movement)		
8	Identify inverter characteristic	▶ Compensating for inverter influence on output voltage	
9	Identify motor parameters	▶ Have motor parameters identified by the i700 servo inverter	
10	Identify Lh saturation characteristic	▶ Asynchronous motor (ASM): Identifying the Lh saturation characteristic	
☑ Write access ☑ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_8

0x2832 | 0x3032 - Identification: Status word

Display of the status of an activated function for automatic parameter identification.

Display area (min. value | unit | max. value)			Initialisation	
0		65535	0	
Value is bit-coded:			Info	
Bit 0	Enable identification		An identification was activated via 0x2825 (or 0x3025 for axis B) with a value > 4.	
Bit 1	Identification in progress		The identification selected via 0x2825 (or 0x3025 for axis B) is carried out (changes to 1 after the controller enable).	
Bit 2	Identification complete		Identification has been completed	
Bit 3	Identification failed		The identification failed or has been aborted by controller inhibit.	
Bits 4-15	Reserved			
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☐ TX				UNSIGNED_16

5.2.1 Enable/inhibit operation via control word

The operating mode can only be selected in 0x2825 (or 0x3025f or axis B) when operation is inhibited (pulse inhibit). In order to start the corresponding procedure after the selection, the operation must be enabled explicitly.

Required steps to enable the operation after an STO:

1. Apply the STO terminals of the axis to 24 V.
2. Use the control word (0x6040 or 0x6840 for axis B) to execute the following commands one after another:

Command	Setting in the control word	Notes
Shutdown	6	A changeover from the "Switch on disabled" device status to the "Ready to switch on" device status takes place.
Switch on	7	The switch-on inhibit which is active after switch-on or reset of an error (acknowledgement) is deactivated. A changeover to the "Switched on" device status takes place.
Enable operation	15	The operation is enabled and an active quick stop is quit again. A changeover to the "Operation enabled" device status takes place.

Access to the control word via SDO access

The control word is part of the standard mapping and is preset in the cyclic interface. In this configuration, the control word commands have to be initiated by the controller via PDO access. If this is not possible, for instance, since a manual intervention through the controller is not intended, it is possible to switch off the control via PDO and carry out the enable via an SDO access:

1. Set the object 0x2824 (or 0x3024 for axis B) to "0: Off" to switch off the control via PDO.
2. An SDO access to the control word (0x6040 or 0x6840 for axis B) serves to execute the commands given above one after another.
3. After having executed the commands in the object 0x2824 (or 0x3024 for axis B), select "1: Activate" to re-activate the control via PDO.

Inhibit operation via control word (pulse inhibit)

In order to inhibit the enabled operation without using the STO terminals, use the control word (0x6040 or 0x6840 for axis B) to execute the following command:

Command	Setting in the control word	Notes
Disable operation	7	The pulse inhibit is set. When the automatic brake operation is activated, it is waited for the parameterised brake closing time until the brake is applied before the pulse inhibit is set. It is changed back to the "Switched on" device status.

In this case, a subsequent renewed enable only requires (without using the STO terminals) the control word command "Enable operation" (setting "15").

5.2.2 Saving changed parameters safe against mains failure

If control parameters are changed during the commissioning phase, e.g. by the functions for automatic parameter identification, the changed control parameters must be uploaded from the i700 servo inverter into the controller for permanent storage.

Note!

Currently, the changed controller parameters cannot be transferred directly into the object list of the corresponding axis using the »PLC Designer« but only indirectly via the »EASY Starter« (see the following instructions).

How to transfer parameters changed in the i700 servo inverter into the PLC project:

In the »EASY Starter«:

1.  Upload parameter set and save as Lenze parameter file (*.gdc).

In the »PLC Designer«:

2. Go to the *Device view* and select the corresponding i700 servo inverter.
3. Go to the **Project** menu and select the **Device parameters→Import device parameters** command to import the Lenze parameter file (*.gdc) into the selected device.
4. Save PLC project.

5.3 Wiring check by means of manual test modes

Before the parameterisation of the actual control is started, the wiring of the motor (power and encoder connection) should be checked for errors and, if required, should be corrected.

- For this purpose, the following manual test modes can be activated via object 0x2825 (or 0x3025 for axis B) when the controller is inhibited:
  - Manual test mode "voltage/frequency"
  - Manual test mode "current/frequency"
- The parameters for the test modes can be adapted via object 0x2835 (or 0x3035 for axis B). For this, observe the notes in the description of the respective test mode.

0x2835 | 0x3035 - Manual test mode: Settings

Sub.	Name	Lenze setting	Data type
1	Manual test mode: Current setpoint	0 %	INTEGER_16
2	Manual test mode: Frequency	0.0 Hz	INTEGER_16
3	Manual test mode: Starting angle	0.0 °	INTEGER_16

Subindex 1: **Manual test mode: Setpoint current**				
Selection of the r.m.s. value of a phase current for test mode • 100 % ≡ rated motor current (0x6075 or 0x6875 for axis B)				
Setting range (min. value | unit | max. value)			Lenze setting	
0	%	1000	0 %	
☑ Write access ☐ CINH ☑ OSC ☐ P ☑ RX ☐ TX				INTEGER_16

Subindex 2: **Manual test mode: Frequency**				
Selection of the frequency for test mode				
Setting range (min. value | unit | max. value)			Lenze setting	
-1000.0	Hz	1000.0	0.0 Hz	
☑ Write access ☐ CINH ☑ OSC ☐ P ☑ RX ☐ TX			Scaling: 1/10	INTEGER_16

Subindex 3: **Manual test mode: Starting angle**				
Selection of the starting angle for test mode **Note!** In the case of the synchronous motor, a jerky compensating movement occurs after controller enable if the pole position of this movement does not correspond to the starting angle.				
Setting range (min. value | unit | max. value)			Lenze setting	
-1000.0	°	1000.0	0.0 °	
☑ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			Scaling: 1/10	INTEGER_16

5.3.1 Manual test mode "voltage/frequency"

Functional description

In this test mode, a rotating field voltage with the set output frequency f_{out} is output at the motor terminals after controller enable.

- If the frequency selection is positive, the motor should rotate clockwise when one is looking at the A-side of the motor. If this is not the case, there is a simple rotating movement in the motor phases.
- The level of the output voltage is determined via the following equation:

Equation for calculating the output voltage

$$U_{out} = f_{out} \cdot \frac{U_{rated}}{f_{rated}}$$

Axis A	Axis B	Symbol	Description	Dimension unit
0x2D82	0x3582	V_{out}	Current output voltage	V
0x2835:2	0x3035:2	f_{out}	Output frequency for test mode	Hz
0x2B01:1	0x3301:1	V_{rated}	V/f rated voltage	V
0x2B01:2	0x3301:2	f_{rated}	V/f base frequency	Hz
Greyed out = read access only				

By means of the manual "Voltage/frequency" test mode, the wiring of the feedback system can also be checked.

- If the feedback system is set correctly, the following actual speed should be shown:

Equation for calculating the actual speed

$$n_{act} = \frac{f_{out}}{zp_{motor}} \cdot 60$$

Axis A	Axis B	Symbol	Description	Dimension unit
0x606C	0x686C	n_{act}	Velocity actual value	rpm
0x2835:3	0x3035:3	f_{out}	Output frequency for test mode	Hz
0x2C01:1	0x3401:1	zp_{Motor}	Motor - number of pole pairs	
Greyed out = read access only				

Preconditions for the execution

- The motor must be able to rotate freely.
- The controller is free of errors and is in the "Switched on" device status.

Response of the motor during the execution

The motor rotates as a function of the set output frequency.

How to activate the manual test mode "voltage/frequency":

1. If the controller is enabled, inhibit the controller.
   ▸ Enable/inhibit operation via control word (📖 59)
2. Set object 0x2825 (or 0x3025 for axis B) to "1" to change to the "Voltage/frequency" test mode.
3. Enable the controller to start the test mode.
4. To stop the test mode:
   - Inhibit controller.
   - Set object 0x2825 (or 0x3025 for axis B) to "0" to change back to the CiA402 mode.

5.3.2 Manual test mode "current/frequency"

Functional description

In this test mode, three phase currents are injected in the connected motor after controller enable.

- Via the following objects, the test mode can be adapted:

Object		Info	Data type
Axis A	Axis B		
0x2835:1	0x3035:1	r.m.s. value of a phase current • Selection in [%] relative to the rated motor current.	INTEGER_16
0x2835:2	0x3035:2	Frequency	INTEGER_16
0x2835:3	0x3035:3	Starting angle	INTEGER_16

- The current phase currents can be read via the following objects:

Object		Info	Data type
Axis A	Axis B		
0x2D83:1	0x3583:1	Motor current zero system	INTEGER_32
0x2D83:2	0x3583:2	Motor current phase U	INTEGER_32
0x2D83:3	0x3583:3	Motor current phase V	INTEGER_32
0x2D83:4	0x3583:4	Motor current phase W	INTEGER_32
Greyed out = read access only			

Advantages compared to the manual test mode "voltage/frequency":

- The current does not set freely, but is controlled to a defined value.
- If a synchronous motor is connected, the generated torque can be predicted.

Preconditions for the execution

- The motor must be able to rotate freely.
- The controller is free of errors and is in the "Switched on" device status.

Response of the motor during the execution

The motor rotates as a function of the set output frequency.

Stop!

In the case of the synchronous motor, a jerky compensating movement occurs after controller enable if the pole position of this movement does not correspond to the starting angle.

How to activate the manual test mode "current/frequency":

1. If the controller is enabled, inhibit the controller.
   ▸ Enable/inhibit operation via control word (📖 59)
2. Set object 0x2825 (or 0x3025 for axis B) to "2" to change to the "Current/frequency" test mode.
3. Enable the controller to start the test mode.
4. To stop the test mode:
   - Inhibit controller.
   - Set object 0x2825 (or 0x3025 for axis B) to "0" to change back to the CiA402 mode.

5.4 Manual control

As an alternative wiring check, the "Manual control" mode can be activated.

Functional description

The "manual control" mode enables manual operation of the i700 servo inverter via the »EASY Starter« PC tool .

- In the manual control mode both the current and the frequency are run to the set final value via a parameterisable ramp time. The ramp times constitute the time span during which the respective parameter is run from zero to the final value.
- In the manual control mode holding brakes, if any, are automatically released. ▶ Holding brake control (📖 157)

Note!

Controlled synchronous motors (SM) can only be traversed under various restrictions:

Restriction 1: The speed variation per control cycle may only be very minor.

- Either make minor changes/leaps of the "Manual jog: Frequency" parameter or select the "Manual jog: Ramp time - frequency" very high.

Restriction 2: The "Manual jog: Frequency" parameter must be so low that the voltage range is not left or that the field weakening range is not reached.

- Provide a reserve of approximately 20 % towards the voltage limit. The voltage requirement is proportional to the speed and thus proportional to the frequency; the voltage limit is determined by the DC-bus voltage.
- For 400-V motors in combination with a 400-V mains, "voltage range" roughly describes the range below the rated motor speed.

Restriction 3: The "Manual jog: Setpoint current" parameter must be selected so high that the friction and load torques of the machine at final speed (= proportional to the parameter "Manual jog: Frequency") can be overcome.

Comparison of the test mode "current/frequency" and the "manual control" mode

The following table shows the differences between the two modes:

<table>
<tr><th colspan="3">Test mode "Current/frequency"</th><th colspan="3">Mode "Manual control"</th></tr>
<tr><td colspan="3">Current setpoint is pending immediately after controller enable.
• 100 % ≡ rated motor current (0x6075 or 0x6875 for axis B)</td><td colspan="3">Current setpoint is run to the final value via a parameterisable ramp time after controller enable.
• 100 % ≡ rated axis current (0x2DDF:1 or 0x35DF:1 for axis B)</td></tr>
<tr><td>Axis A</td><td>Axis B</td><td>Parameter</td><td>Axis A</td><td>Axis B</td><td>Parameter</td></tr>
<tr><td>0x2835:1</td><td>0x3035:1</td><td>Current setpoint</td><td>0x2836:1</td><td>0x3036:1</td><td>Current setpoint (final value)</td></tr>
<tr><td>-</td><td>-</td><td>(No ramp can be set)</td><td>0x2836:3</td><td>0x3036:3</td><td>Ramp time</td></tr>
<tr><td colspan="3">Test frequency is pending immediately after controller enable.</td><td colspan="3">Test frequency is run up to the final value via a parameterisable ramp time after controller enable (and is also run down again). The frequency ramp starts when the current setpoint has reached its final value.</td></tr>
<tr><td>Axis A</td><td>Axis B</td><td>Parameter</td><td>Axis A</td><td>Axis B</td><td>Parameter</td></tr>
<tr><td>0x2835:2</td><td>0x3035:2</td><td>Test frequency</td><td>0x2836:2</td><td>0x3036:2</td><td>Test frequency (final value)</td></tr>
<tr><td>-</td><td>-</td><td>(No ramp can be set)</td><td>0x2836:4</td><td>0x3036:4</td><td>Ramp time</td></tr>
<tr><td colspan="3">Starting angle is pending immediately after controller enable or the rotating field starts with the starting angle.
• In the case of the synchronous motor, a jerky compensating movement occurs once if the starting angle does not correspond to the current pole position of the synchronous motor.</td><td colspan="3" rowspan="3">Starting angle is the current commutation angle.
• For the synchronous motor, therefore no compensating movement should be effected.</td></tr>
<tr><td>Axis A</td><td>Axis B</td><td>Parameter</td></tr>
<tr><td>0x2835:3</td><td>0x3035:3</td><td>Starting angle</td></tr>
<tr><td colspan="3" rowspan="3">Duration unlimited.
(If permitted by the device utilisation and motor temperature.)</td><td colspan="3">Duration adjustable.
The time starts with the start of the frequency ramp (when the current setpoint has run up to its final value).
• If the test frequency is not rewritten within the time set for the time monitoring, the frequency is decreased to zero via the parameterised ramp.
• When 0 Hz has been reached, the controller changes to the error status and is inhibited. After the error has been acknowledged, the "Switch on disabled" device status has to be changed to the "Operation enabled" device status again before manual control can be continued. ▸ Enable/inhibit operation via control word</td></tr>
<tr><td>Axis A</td><td>Axis B</td><td>Parameter</td></tr>
<tr><td>0x2836:5</td><td>0x3036:5</td><td>Time span for time monitoring</td></tr>
</table>

Preconditions for the execution

- The motor must be able to rotate freely.
- The controller is free of errors and is in the "Switched on" device status.

Response of the motor during the execution

The motor rotates according to the manual jog commands.

How to activate the manual control:

1. If the controller is enabled, inhibit the controller.
   ▶ Enable/inhibit operation via control word (📖 59)
2. Set object 0x2825 (or 0x3025 for axis B) to "4" to change to the "Manual control" mode.
3. Enable the controller to start the manual control.
   - In the »EASY Starter«, the controller can be enabled via the operator dialog for manual control. The operator dialog also takes priority over the control word (0x6040 or 0x6840 for axis B).
   - The possible settings for the manual control mode are described in the following.
4. To stop the manual control:
   - Inhibit controller.
   - Set object 0x2825 (or 0x3025 for axis B) to "0" to change back to the CiA402 mode.

0x2836 | 0x3036 - Manual jog: Settings

Sub.	Name	Lenze setting	Data type
▶ 1	Manual jog: Current setpoint	30 %	UNSIGNED_16
▶ 2	Manual jog: Frequency	0.0 Hz	INTEGER_16
▶ 3	Manual jog: Ramp time - current	0 ms	UNSIGNED_16
▶ 4	Manual jog: Ramp time - frequency	500 ms	UNSIGNED_16
▶ 5	Manual jog: Time monitoring	2500 ms	UNSIGNED_32
▶ 6	Manual jog: Current controller - gain	20.00 V/A	UNSIGNED_32
▶ 7	Manual jog: Current controller - reset time	20.00 ms	UNSIGNED_32

Subindex 1: **Manual jog: Setpoint current**				
Selection of the current setpoint for the manual control • 100 % ≡ rated axis current (0x2DDF:1 or 0x35DF:1 for axis B)				
Setting range (min. value | unit | max. value)			Lenze setting	
0	%	200	30 %	
☑ Write access ☐ CINH ☐ OSC ☐ P ☑ RX ☐ TX				UNSIGNED_16

Subindex 2: **Manual jog: Frequency**				
Selection of the frequency for manual control				
Setting range (min. value | unit | max. value)			Lenze setting	
-1000.0	Hz	1000.0	0.0 Hz	
☑ Write access ☐ CINH ☑ OSC ☐ P ☑ RX ☐ TX			Scaling: 1/10	INTEGER_16

Subindex 3: Manual jog: Ramp time - current				
Time span during which the current setpoint is run from zero to the final value set.				
Setting range (min. value | unit | max. value)			Lenze setting	
0	ms	1000	0 ms	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

Subindex 4: Manual jog: Ramp time - frequency				
Time span during which the frequency is run from zero to the final value set.				
Setting range (min. value | unit | max. value)			Lenze setting	
0	ms	10000	500 ms	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

Subindex 5: Manual jog: Time monitoring				
Time span for time monitoring • The manual control features a time monitoring function which is coupled to a write access to subindex 2 (Manual jog: Frequency). • If no write access to the subindex 2 takes place within the time period set here, the frequency is lead to zero via the parameterised ramp. When the 0 Hz have been reached, the inverter changes to the error status and is inhibited. After acknowledging the error, switch the CiA402 state machine from the "switch on inhibited" back to the "operation enabled" status before proceeding with manual jog.				
Setting range (min. value | unit | max. value)			Lenze setting	
0	ms	100000	2500 ms	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_32

Subindex 6: Manual jog: Current controller gain				
Setting range (min. value | unit | max. value)			Lenze setting	
0.00	V/A	750.00	20.00 V/A	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/100	UNSIGNED_32

Subindex 7: Manual jog: Current controller reset time				
Setting range (min. value | unit | max. value)			Lenze setting	
0.01	ms	2000.00	20.00 ms	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/100	UNSIGNED_32

5.5 Setting the control mode

The i700 servo inverter supports different modes for open-loop/closed-loop motor control:

- Servo control for synchronous motor (SM)
- Servo control for asynchronous motor (ASM)
- V/f characteristic control for asynchronous motor (ASM)

Servo control

The field-oriented servo control is based on a decoupled, separated control of the torque-producing and field-producing current component. The motor control is based on a fed back, field-oriented and cascaded controller structure and enables dynamic and stable operation in all four quadrants. It can be used for synchronous motors (SM) and asynchronous motors (ASM).

V/f characteristic control

The V/f characteristic control is a motor control mode for typical frequency inverter applications on the basis of a simple and robust control mode for the operation of asynchronous motors with linear or square-law load torque characteristics (e.g. fans). This motor control mode is also suitable for group drives and special motors.

0x2C00 | 0x3400 - Motor control

Setting of the mode for open-loop/closed loop motor control

Selection list (Lenze setting printed in bold)		Info	
1	**Servo control - synchronous motor (SM)**	Servo control for synchronous motor (SM)	
2	Servo control - asynchronous motor (ASM)	Servo control for asynchronous motor (ASM)	
3	Reserved	From version 01.03	
6	VFC: V/f characteristic control	V/f characteristic control for asynchronous motor (ASM)	
☑ Write access ☑ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_8

5.6 Accepting/adapting plant parameters

The "plant parameters" summarise all parameters which result from the combination of motor and load. These characterise the transfer behaviour of the entire controlled system including the required monitoring modes.

- The plant parameters depend on the application in which the controller and motor are used.
- When a Lenze motor is selected, respective plant parameters are suggested for the load-free operation. This enables the motor to be moved without any load. During operation with a load it is required to change the plant parameters according to the load. Details on how to determine the adapted plant parameters are described in the corresponding objects.

Overview of plant parameters

Object		Info	Lenze setting		Control	
Axis A	Axis B		Value	Unit	Servo	V/f
0x6080	0x6880	Max motor speed	6075	r/min	●	●
0x6073	0x6873	Max current	150	%	●	●
0x2900:1	0x3100:1	Speed controller: Gain	0.00038	Nm/rpm	●	
0x2900:2	0x3100:2	Speed controller: Reset time	15.0	ms	●	
0x2900:3	0x3100:3	Speed controller: Rate time	0.00	ms	●	
0x2904	0x3104	Speed: Actual speed - filter time	0.6	ms	●	●
0x2D44:1	0x3544:1	Motor speed monitoring: Threshold	8000	r/min	●	●

5.7 Compensating for inverter influence on output voltage

Note!

In the V/f characteristic control mode the procedure described below is recommended because a well-adapted inverter characteristic leads to a significantly improved drive behaviour during V/f operation.

Functional description

An inverter generates a three-phase voltage system with pulse width modulation. Inherent to its functional principle, the inverter also generates current-dependent and switching frequency-dependent losses which influence the output voltage. The motor voltage actually provided at the output terminals is not measured in order to compensate for deviations if need be. Instead, an adjustment is made by means of the inverter characteristic to compensate for deviations.

Among other things, the inverter characteristic depends on the length of the motor cable and at least has to be individually determined once for the connected motor by means of the device command "Calculate inv. characteristic". For an automatic determination of the motor parameters, this ensures that the current has a sinusoidal form.

Danger!

This procedure may only be carried out during commissioning, not during operation!

During the procedure the motor is energised so that:
- it cannot be excluded that the connected mechanical components may move!
- the windings heat up.
  If you repeat the procedure, ensure that the motor is not thermally overloaded (particularly if no temperature feedback is used).

Identification of the inverter characteristic - procedure

If no error is pending, the motor is energised during the procedure with a maximum DC current corresponding to the lower of the two following values:

$$\sqrt{2} \cdot \text{Rated device current}$$

or

$$\sqrt{2} \cdot 1.8 \cdot \text{Rated motor current}$$

- Ideally, the first value should be reached, the second value is to ensure that the load on the motor is not too high during this test.
- During the procedure, the motor current increases to the maximum value specified and falls back to "0" to repeat the cycle with a negative current sign. Altogether, the maximum value is reached four times.
- The switching frequency of the inverter is set to rated switching frequency and after the procedure, it is reset to the original value.
  - If the switching frequency should be changed later during operation, the characteristic will automatically be adapted to the current switching frequency.

Preconditions for the execution

- The motor may be firmly braked.
- The controller is free of errors and is in the "Switched on" device status.

Response of the motor during the execution

If the motor is not firmly braked, it will move slightly.

How to determine the inverter characteristic:

1. If the controller is enabled, inhibit the controller.
   ▶ Enable/inhibit operation via control word (📖 59)
2. Set object 0x2825 (or 0x3025 for axis B) to "8" to change to the "Inverter characteristic: Identification" operating mode.
3. Enable the controller to start the procedure.

   Note: By means of controller inhibit, the started procedure can be cancelled any time. Characteristic values that have already been determined are rejected in this case.

After successful completion...

...the controller will be inhibited automatically and the points of the determined inverter characteristic will be set in object 0x2947 (or 0x3147 for axis B).

- For permanent storage, the changed settings must be uploaded to the controller from the i700 servo inverter.

  The »EASY Starter« serves to save the parameter settings of the i700 servo inverter as parameter file (*.gdc). In the »PLC Designer«, this file can then be imported in the corresponding axis.
  ▶ Saving changed parameters safe against mains failure (📖 60)
- The inverter characteristic must only be detected again if the controller, motor, or motor cable has changed e.g. due to an exchange.
- The controller inhibit automatically set by the procedure can be deactivated via the Controlword (0x6040 or 0x6840 for axis B) (setting = 7, 15).

In the event of an error

If an error occurs during the procedure or the pulse inhibit gets active (e.g. due to short-time undervoltage), the procedure is terminated with controller inhibit without a change in settings.

Advanced settings

For characteristic detection, the current controller is parameterised automatically at the beginning of the identification process. For motors with a very low stator leakage inductance (< 1 mH), an automatic parameter setting may fail and the actual identification process is aborted with an error message like e.g. "short circuit".

- For such a case, it is possible to set the current controller manually via the object 0x2942 (or 0x3142 for axis B).
- Whether the current controller is to be selected automatically or the values below 0x2942 (or 0x3142 for axis B) are active, is selected via the object 0x2DE0:1 (or 0x35E0:1 for axis B).

0x2DE0 | 0x35E0 - Advanced settings

Sub.	Name	Lenze setting	Data type
▸ 1	Current controller: Setting for identification	0: Automatic setting	UNSIGNED_8
▸ 2	Sensorless SM: Injection signal during test mode	0: Off	UNSIGNED_8

Subindex 1: **Current controller: Setting for identification**			
Selection list (Lenze setting printed in bold)			
0	**Automatic setting**		
1	Manual setting (0x2942/0x3142)		
☑ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			UNSIGNED_8

Subindex 2: **Sensorless SM: Injection signal during test mode**			
Selection list (Lenze setting printed in bold)			
0	**Off**		
1	On		
☑ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			UNSIGNED_8

Loading the standard inverter characteristic

If an incorrect inverter characteristic has been determined or none at all, it is possible to load a device-typical standard inverter characteristic.

How to load the standard inverter characteristic:

1. Set object 0x2822 (or 0x3022 for axis B) to "1".
   - The progress of the procedure is shown in object 0x2823 (or 0x3023 for axis B).
2. For permanent storage: After completion of the procedure, upload the inverter characteristic set in 0x2947 (or 0x3147 for axis B) to the controller from the i700 servo inverter.

   The »EASY Starter« serves to save the parameter settings of the i700 servo inverter as parameter file (*.gdc). In the »PLC Designer«, this file can then be imported in the corresponding axis. ▸ Saving changed parameters safe against mains failure (📖 60)

0x2947 | 0x3147 - Inverter characteristic: Voltage grid points (y)

y values of the 17 grid points of the inverter characteristic

Sub.	Name	Lenze setting	Data type
1	IC: y1 = U01 (x = 0.00 %)	0.00 V	UNSIGNED_16
2	IC: y2 = U02 (x = 6.25 %)	0.00 V	UNSIGNED_16
3	IC: y3 = U03 (x = 12.50 %)	0.00 V	UNSIGNED_16
4	IC: y4 = U04 (x = 18.75 %)	0.00 V	UNSIGNED_16
5	IC: y5 = U05 (x = 25.00 %)	0.00 V	UNSIGNED_16
6	IC: y6 = U06 (x = 31.25 %)	0.00 V	UNSIGNED_16
7	IC: y7 = U07 (x = 37.50 %)	0.00 V	UNSIGNED_16
8	IC: y8 = U08 (x = 42.75 %)	0.00 V	UNSIGNED_16
9	IC: y9 = U09 (x = 50.00 %)	0.00 V	UNSIGNED_16
10	IC: y10 = U10 (x = 56.25 %)	0.00 V	UNSIGNED_16
11	IC: y11 = U11 (x = 62.50 %)	0.00 V	UNSIGNED_16
12	IC: y12 = U12 (x = 68.75 %)	0.00 V	UNSIGNED_16
13	IC: y13 = U13 (x = 75.00 %)	0.00 V	UNSIGNED_16
14	IC: y14 = U14 (x = 81.25 %)	0.00 V	UNSIGNED_16
15	IC: y15 = U15 (x = 87.50 %)	0.00 V	UNSIGNED_16
16	IC: y16 = U16 (x = 93.25 %)	0.00 V	UNSIGNED_16
17	IC: y17 = U17 (x = 100.00 %)	0.00 V	UNSIGNED_16
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			

5.8 Setting the motor parameters for the servo control

If the servo control (SM, ASM) is applied, it is required to parameterise the so-called motor model first. These are electrical variables which are provided in the i700 servo inverter as parameters.

5.8.1 Lenze motor: Easy loading of motor data from the catalogue via the »PLC-Designer«

For Lenze motors with a standard feedback system installed by Lenze, the required motor parameters are provided by the »PLC Designer« during the motor selection by means of catalogues and automatically copied to the i700 servo inverter in the default setting.

In this case, you can directly go to the next commissioning step:
▸ Setting the feedback system for the servo control

5.8.2 Motors of other manufacturers or no catalogue data available: Three possibilities to create the setting

For motors of other manufacturers or if there are no catalogue data available for the motor, three alternative possibilities for setting the motor parameters are provided:

A. Enter motor nameplate data
(and have motor model parameters determined automatically)

B. Set motor parameters manually

C. Determine motor parameters automatically via "motor parameter identification"

The following subchapters provide detailed information on the three alternative methods. The Engineering tool saves the used method to the following object so that the suitable parameterisation dialogs are available the next time an online connection is present.

0x2C08 | 0x3408 - motor: Setting method - motor parameters

Selection list (Lenze setting printed in bold)		Info	
1	**Select from catalogue (Lenze motor)**	For Lenze motors with a standard feedback system installed by Lenze, the required motor parameters are provided by the »PLC Designer« during the motor selection by means of catalogues and automatically copied to the i700 servo inverter in the default setting.	
2	Enter motor nameplate data (other motors)	▸ Enter motor nameplate data (and have motor model parameters determined automatically)	
3	Manually (other motors)	▸ Set motor parameters manually	
4	Identification in progress (all motors)	▸ Determine motor parameters automatically via "motor parameter identification"	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_8

5.8.2.1 Enter motor nameplate data

If the equivalent circuit data of the motor required for the motor model are not known, they can automatically be determined by approximation by the i700 servo inverter by means of the motor nameplate data before they are set.

How to have the equivalent circuit data determined by the i700 servo inverter:

1. Set the complete motor nameplate data in object 0x2C01 (or 0x3401 for axis).
2. Set the rated motor current in object 0x6075 (or 0x6875 for axis B).
3. Set object 0x2822 (or 0x3022 for axis B) to "3" in order to have the equivalent circuit data determined by approximation.
   - The progress of the procedure is shown in object 0x2823 (or 0x3023 for axis B).
4. For permanent storage: Upload the set equivalent circuit data to the controller from the i700 servo inverter after the procedure has been completed:
   - For asynchronous motor (ASM): Object 0x2C02 (or 0x3402 for axis B).
   - For synchronous motor (SM): Object 0x2C03 (or 0x3403 for axis B).

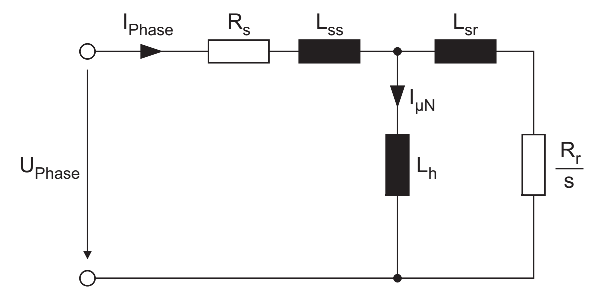
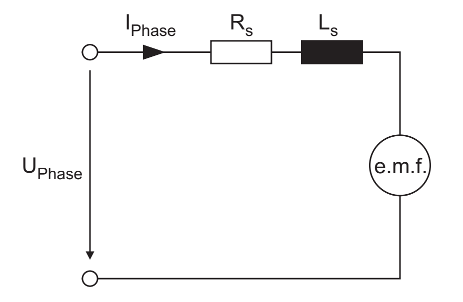
   The »EASY Starter« serves to save the parameter settings of the i700 servo inverter as parameter file (*.gdc). In the »PLC Designer«, this file can then be imported in the corresponding axis. ▶ Saving changed parameters safe against mains failure (📖 60)

5.8.2.2 Set motor parameters manually

If all required motor data are known (e.g. by means of a data sheet provided by the motor manufacturer), they can be set manually in the following parameters:

Object		Name	Required for	
Axis A	Axis B		SM	ASM
0x2C01:1	0x3401:1	Motor: Number of pole pairs	●	●
0x2C01:2	0x3401:2	Motor: Stator resistance (value at 20°C)	●	●
0x2C01:3	0x3401:3	Motor: Stator leakage inductance	●	●
0x2C01:4	0x3401:4	Motor: Rated speed	●	●
0x2C01:5	0x3401:5	Motor: Rated frequency	●	●
0x2C01:6	0x3401:6	Motor: Rated power	●	●
0x2C01:7	0x3401:7	Motor: Rated voltage	●	●
0x2C01:8	0x3401:8	Motor: Rated cosine phi	●	●
0x2C01:9	0x3401:9	Motor: Insulation class	●	●
0x2C02:1	0x3402:1	Motor (ASM): Rotor resistance (value at 20 °C)		●
0x2C02:2	0x3402:2	Motor (ASM): Mutual inductance		●
0x2C02:3	0x3402:3	Motor (ASM): Magnetising current		●
0x2C03:1	0x3403:1	Motor (SM): e.m.f. constant (KELL)	●	
0x2C03:2	0x3403:2	Motor (SM): Pole position	●	
0x2C03:3	0x3403:3	Motor (SM): Temperature coefficient - magnets	●	
0x6075	0x6875	Motor rated current	●	●
0x6076	0x6876	Motor rated torque	●	●
Greyed out = read access only				

Motor equivalent circuit

Async. motor	Synchronous motor
	

Axis A	Axis B	Symbol	Description	Dimension unit
0x2C01:2	0x3401:2	R_s	Motor stator resistance (value at 20°C)	Ω
0x2C01:3	0x3401:3	L_{ss}	Motor stator leakage inductance (ASM)	H
		L_s	Motor stator inductance (SM)	H
-	-	L_{sr}	Motor rotor leakage inductance (is assumed to be equal to L_{ss})	H
0x2C02:1	0x3402:1	R_r	Motor rotor resistance (value at 20°C)	Ω
0x2C02:2	0x3402:2	L_h	Mutual motor inductance	H
0x2C02:3	0x3402:3	$I_{\mu N}$	Motor magnetising current	A
-	-	s	Slip	
-	-	e.m.f.	Electromotive force	

5.8.2.3 Determine motor parameters automatically via "motor parameter identification"

Danger!

This procedure may only be carried out during commissioning, not during operation!

- During the procedure the motor is energised so that:
  - it cannot be excluded that the connected mechanical components may move!
  - the windings heat up.
    If you repeat the procedure, ensure that the motor is not thermally overloaded (particularly if no temperature feedback is used).

Functional description

The motor parameters listed in the table below can be determined automatically via the motor parameter identification function if they are not known. The resistance values are converted via the actual motor temperature into values that correspond to a temperature of 20°C. If a thermal detector is not connected, a temperature value of 90°C is assumed.

Object		Name	Required for	
Axis A	Axis B		SM	ASM
0x2C01:2	0x3401:2	Motor: Stator resistance (value at 20°C)	●	●
0x2C01:3	0x3401:3	Motor: Stator leakage inductance (L_{ss})	●	●
0x2C02:1	0x3402:1	Motor (ASM): Rotor resistance (value at 20 °C)		●
0x2C02:2	0x3402:2	Motor (ASM): Mutual inductance (L_h)		●
0x2C02:3	0x3402:3	Motor (ASM): Magnetising current		●

Sequence of the motor parameter identification

The apparent resistance of the plant is determined for approx. 30 different frequencies. Then a mathematical procedure is used to extract the electrical parameters of the motor.

- Since the procedure starts with very low frequencies and always considers several complete periods, the whole process takes approx. 3 minutes.
- During the procedure, the motor is energised with a current, the r.m.s. value of which corresponds to the lower of the following two values:

Rated device current

or

$$\frac{1}{2} \cdot \text{Rated motor current}$$

Tip!

If an asynchronous motor is to be identified, the identification should be executed with half the rated motor current in order to achieve an optimal result. For this purpose, the rated device current has to be higher than half the rated motor current. In case of a synchronous motor, this is irrelevant.

After the parameters have been extracted from the impedance, they are checked for consistency with the required rated values. If an inconsistent parameter set is detected, is this an indication of faulty rated values on the nameplate.

Preconditions for the execution

- The synchronous motor must be able to rotate freely.
- The asynchronous motor may be firmly braked.
- The controller is free of errors and is in the "Switched on" device status.
- The motor parameters listed in the following table are excluded from the automatic determination and must therefore be adapted to the motor used (see motor nameplate before the determination.

Object		Name
Axis A	Axis B	
0x6075	0x6875	Motor rated current (The current amount for the procedure is derived from this specification)
0x2C01:2	0x3401:2	Motor: Stator resistance (Default setting is used as starting value for the automatic determination.)
0x2C01:4	0x3401:4	Motor: Rated speed
0x2C01:5	0x3401:5	Motor: Rated frequency
0x2C01:6	0x3401:6	Motor: Rated power
0x2C01:7	0x3401:7	Motor: Rated voltage

Response of the motor during the execution

- A DC current is superimposed over the identification current that keeps the motor idling. After the controller enable, the shaft will adjust once, which is irrelevant to measurement though.
- With asynchronous motors, slight rotations might possibly occur. Their influence on the measurements is, however, not worth mentioning.

Note!

- In case of uncertainties, the measurement should be repeated several times to check if the results for the stator resistance, the leakage inductance of the stator and the rotor resistance differ widely. This should not be the case.
- The mutual inductance and the cos(φ) values are not that important for the diagnostics, because they are strongly non-linear.

How to carry out the motor parameter identification:

1. If the controller is enabled, inhibit the controller.
   ▶ Enable/inhibit operation via control word (📖 59)
2. Set object 0x2825 (or 0x3025 for axis B) to "9" to change to the "Motor: Parameter identification" operating mode.
3. Enable the controller to start the procedure.

   Note: By means of controller inhibit, the procedure started can be cancelled any time, if required, without a change in settings.

After successful completion...

...the controller is automatically inhibited and the determined motor data are set in the corresponding objects (0x2C01 and 0x2C02 or 0x3401 and 0x3402 for axis B).

- For permanent storage, the changed settings must be uploaded to the controller from the i700 servo inverter.
  The »EASY Starter« serves to save the parameter settings of the i700 servo inverter as parameter file (*.gdc). In the »PLC Designer«, this file can then be imported in the corresponding axis.
  ▶ Saving changed parameters safe against mains failure (📖 60)
- The controller inhibit automatically set by the procedure can be deactivated via the Controlword (0x6040 or 0x6840 for axis B) (setting = 7, 15).

In the event of an error

If an error occurs during the procedure or the pulse inhibit gets active (e.g. due to short-time undervoltage), the procedure is terminated with controller inhibit without a change in settings.

Advanced settings

For motor parameter detection, the current controller is parameterised automatically at the beginning of the identification process. For motors with a very low stator leakage inductance (< 1 mH), an automatic parameter setting may fail and the actual identification process is aborted with an error message like e.g. "short circuit".

- For such a case, it is possible to set the current controller manually via the object 0x2942 (or 0x3142 for axis B).
- Whether the current controller is to be selected automatically or the values below 0x2942 (or 0x3142 for axis B) are active, is selected via the object 0x2DE0:1 (or 0x35E0:1 for axis B).

5.8.3 Motor parameters (object descriptions)

0x2C01 | 0x3401 - Motor: Common parameters

Sub.	Name	Lenze setting	Data type
1	Motor: Number of pole pairs		UNSIGNED_8
2	Motor: Stator resistance	13.5000 ohms	UNSIGNED_32
3	Motor: Stator leakage inductance	51.000 mH	UNSIGNED_32
4	Motor: Rated speed	4050 rpm	UNSIGNED_16
5	Motor: Rated frequency	270.0 Hz	UNSIGNED_16
6	Motor: Rated power	0.25 kW	UNSIGNED_16
7	Motor: Rated voltage	225 V	UNSIGNED_16
8	Motor: Rated cosine phi	0.80	UNSIGNED_16
9	Motor: Insulation class	4: F (cut-off temperature = 155 °C)	UNSIGNED_8
10	Motor: Designation		STRING(50)

Subindex 1: **Motor: Number of pole pairs**				
Display area (min. value | unit | max. value)			Initialisation	
0		255		
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX				UNSIGNED_8

Subindex 2: **Motor: Stator resistance**				
Setting range (min. value | unit | max. value)			Lenze setting	
0.0000	Ohm	125.0000	13.5000 ohms	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10000	UNSIGNED_32

Subindex 3: **Motor: Stator leakage inductance**				
Setting range (min. value | unit | max. value)			Lenze setting	
0.000	mH	500.000	51.000 mH	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/1000	UNSIGNED_32

Subindex 4: **Motor: Rated speed**				
Setting range (min. value | unit | max. value)			Lenze setting	
0	r/min	50000	4050 rpm	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

Subindex 5: **Motor: Rated frequency**				
Setting range (min. value | unit | max. value)			Lenze setting	
0.0	Hz	1000.0	270.0 Hz	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	UNSIGNED_16

Subindex 6: **Motor: Rated power**				
Setting range (min. value | unit | max. value)			Lenze setting	
0.00	kW	655.35	0.25 kW	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/100	UNSIGNED_16

Subindex 7: **Motor: Rated voltage**				
Setting range (min. value | unit | max. value)			Lenze setting	
0	V	65535	225 V	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

Subindex 8: **Motor: Rated cosine phi**				
Setting range (min. value | unit | max. value)			Lenze setting	
0.00		655.35	0.80	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/100	UNSIGNED_16

Subindex 9: **Motor: Insulation class**			
Selection list (Lenze setting printed in bold)			
0	Y (cut-off temperature = 90°C)		
1	A (cut-off temperature = 105 °C)		
2	E (cut-off temperature = 120°C)		
3	B (cut-off temperature = 130°C)		
4	**F (cut-off temperature = 155°C)**		
5	H (cut-off temperature = 180°C)		
6	G (cut-off temperature > 180°C)		
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_8

Subindex 10: **Motor: Designation**		
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX		STRING(50)

0x2C02 | 0x3402 - Motor (ASM): Parameter

Sub.	Name	Lenze setting	Data type
1	Motor (ASM): Rotor resistance	0.0000 Ohm	UNSIGNED_32
2	Motor (ASM): Mutual inductance	0.0 mH	UNSIGNED_32
3	Motor (ASM): Magnetising current	0.00 A	UNSIGNED_16

Subindex 1: **Motor (ASM): Rotor resistance**				
Setting range (min. value | unit | max. value)			Lenze setting	
0.0000	Ohm	214748.3647	0.0000 Ohm	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10000	UNSIGNED_32

Subindex 2: **Motor (ASM): Mutual inductance**				
Setting range (min. value | unit | max. value)			Lenze setting	
0.0	mH	214748364.7	0.0 mH	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	UNSIGNED_32

Subindex 3: **Motor (ASM): Magnetising current**				
Setting range (min. value | unit | max. value)			Lenze setting	
0.00	A	500.00	0.00 A	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/100	UNSIGNED_16

0x2C03 | 0x3403 - Motor (SM): Parameter

Sub.	Name	Lenze setting	Data type
▸ 1	Motor (SM): e.m.f. constant (KELL)	41.8 V/1000 rpm	UNSIGNED_32
▸ 2	Motor (SM): Pole position resolver	-90.0 °	INTEGER_16
▸ 3	Motor (SM): Temperature coefficient - magnets (kTN)	-0.110 %/°C	INTEGER_16
▸ 4	Motor (SM): Pole position encoder	0.0 °	INTEGER_16

Subindex 1: **Motor (SM): e.m.f. constant (KELL)**				
Setting range (min. value | unit | max. value)			Lenze setting	
0.0	V/1000 rpm	100000.0	41.8 V/1000 rpm	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	UNSIGNED_32

Subindex 2: **Motor (SM): Pole position resolver**				
Setting range (min. value | unit | max. value)			Lenze setting	
-179.9	°	179.9	-90.0 °	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	INTEGER_16

Subindex 3: **Motor (SM): Temperature coefficient - magnets (kTN)**				
Setting range (min. value | unit | max. value)			Lenze setting	
-1.000	%/°C	0.000	-0.110 %/°C	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/1000	INTEGER_16

Subindex 4: **Motor (SM): Pole position encoder**				
From version 01.03				
Setting range (min. value | unit | max. value)			Lenze setting	
-179.9	°	179.9	0.0 °	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	INTEGER_16

0x6075 | 0x6875 - Motor rated current

The rated motor current set here serves as a reference value for the following objects:
- 0x6073 | 0x6873 - Device: Max. current
- 0x6078 | 0x6878 - Current actual value
- 0x2835 | 0x3035 - Manual test mode: Settings
- 0x2C61 | 0x3461 - Pole position identification PPI (360°)
- 0x2C62 | 0x3462 - Pole position identification PPI (min. movement)
- 0x2D4D | 0x354D - Motor utilisation (I^2xt): User-definable characteristic

Setting range (min. value | unit | max. value)			Lenze setting	
0.001	A	500.000	1.300 A	
☑ Write access ☑ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/1000	UNSIGNED_32

0x6076 | 0x6876 - Motor rated torque

The rated motor torque set here serves as a reference value for the following objects:
- 0x6071 | 0x6872 - Target torque
- 0x6072 | 0x6872 - Max. torque
- 0x6074 | 0x6874 - Torque demand
- 0x6077 | 0x6877 - Torque actual value
- 0x60B2 | 0x68B2 - Torque offset
- 0x60E0 | 0x68E0 - Positive torque limit value
- 0x60E1 | 0x68E1 - Negative torque limit value
- 0x2DD6 | 0x35D6 - Torque: Filter cascade

Setting range (min. value | unit | max. value)			Lenze setting	
0.001	Nm	1000.000	0.600 Nm	
☑ Write access ☑ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/1000	UNSIGNED_32

5.9 Setting the feedback system for the servo control

After setting the motor parameters, the feedback system for the servo control must be set.

Note!

The feedback system has already been preselected by the hardware of the available device version. Either the objects for resolver evaluation or the objects for encoder evaluation are effective. Access to ineffective objects of hardware not available is ignored.

The following table shows which parameters are valid for which feedback system:

Object Axis A	 Axis B	Name
▸ General settings		
0x2C45	0x3445	Open circuit in feedback system: Response
0x2C46	0x3446	Feedback system: Specifiable number of revolutions
▸ Settings for "resolver" version		
0x2C03:2	0x3403:2	Motor (SM): Pole position resolver
0x2C43	0x3443	Resolver: Number of pole pairs
▸ Settings for "encoder" version		
0x2C03:4	0x3403:4	Motor (SM): Pole position encoder
0x2C40	0x3440	Encoder: Type
0x2C42:1	0x3442:1	Encoder: Increments / revolution
0x2C42:2	0x3442:2	Encoder: Supply voltage
0x2C42:3	0x3442:3	Encoder: Angle drift - Actual angle error
0x2C42:4	0x3442:4	Encoder: Signal quality - Actual amplitude
▸ Additional settings for SinCos absolute value encoders with HIPERFACE® protocol		
0x2C41:1	0x3441:1	Hiperface: Determined type code
0x2C41:2	0x3441:2	Hiperface: User def. encoder - type code
0x2C41:3	0x3441:3	Hiperface: User def. encoder - specifiable revolutions
0x2C41:4	0x3441:4	Hiperface absolute value fault: Response
0x2C41:5	0x3441:5	Hiperface: Serial number
0x2C41:6	0x3441:6	Hiperface: Raw data - Actual position
0x2C41:7	0x3441:7	Hiperface: Detected Increments / revolution
0x2C41:8	0x3441:8	Hiperface: Type code supported by firmware
0x2C41:9	0x3441:9	Hiperface: Encoder type
0x2C41:10	0x3441:10	Hiperface: Period length linear encoder
Greyed out = read access only		

5.9.1 General settings

Encoder open-circuit monitoring

In the Lenze setting the resolver/encoder cable is monitored for open circuit.

Danger!

If the encoder/resolver is used as motor encoder:
Safe operation of the motor is no longer ensured in the event of an error!

- For safety reasons, always select "Fault" (Lenze setting) as a response for the (open-circuit) monitoring of the encoder/resolver!
- In order to prevent interference injections when an encoder is used, only use shielded motor and encoder cables!

When does the open-circuit monitoring system respond?

Resolver	Multi encoder
• If there is an open circuit in the encoder cable. • If the impedance of the resolver is too high. • In the case of interference injections (EMC interferences).	• If there is an open circuit in the encoder cable.

0x2C45 | 0x3445 - open circuit in the feedback system: Response

If there is an open circuit in the encoder cable, the response set here is triggered.

Selection list (Lenze setting printed in bold)		Info	
0	No response		
1	**Fault**	• In the device statuses "not ready to start" and "switch-on inhibited", a warning is output. • In all other device statuses, the error response set here is executed.	
2	Warning		
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_8

Display parameter

0x2C46 | 0x3446 - feedback system: Specifiable number of revolutions

From version 01.03
Is set by the firmware according to the available version:
0: No absolute value encoder (sin/cos encoder) or resolver with number of pole pairs > 1
1: Hiperface encoder SingleTurn or resolver with number of pole pairs = 1
>1: Hiperface encoder MultiTurn

Display area (min. value | unit | max. value)			Initialisation	
0		65535		
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX				UNSIGNED_16

0x2C5F | 0x345F - Feedback system: Parameter CRC

From version 01.03

Display area (min. value | unit | max. value)			Initialisation	
0		4294967295		
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX				UNSIGNED_32

5.9.2 Settings for "resolver" version

0x2C43 | 0x3443 - Resolver: Number of pole pairs

Setting range (min. value | unit | max. value)			Lenze setting	
1		10	1	
☑ Write access ☑ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_8

Note!

Resolvers with a number of pole pairs > 1 are no absolute value encoders.

- Thus, bit 4 in the Lenze status word 2 (0x2833 or 0x3033 for axis B) remains set to "0".
- The "distinguishable revolutions" specification in 0x2C46 (or 0x3446 for axis B) is also set to "0".

The following applies to synchronous motors:

- In case of integer ratios of the number of pole pairs of the motor (0x2C01:1 or 0x3401:1 for axis B) to the number of pole pairs of the resolver, the pole position identification is only required once.
- In case of non-integer ratios, a pole position identification has to be executed after every 24-V switching operation of the i700 servo inverter.

▶ Synchronous motor (SM): Pole position identification (📖 95)

Resolver error compensation, resolover error identification

The actual position determined via the resolver does not exactly correspond to the actual physical position. There will always be some greater or lesser deviation. The resolver error characteristic shows the deviation as a function of the rotor position. It usually behaves in a typical way due to the following causes:

Cause	Remedy
The inductances of the sine and cosine track of the resolver have slightly different values.	Adaptation of the gains for the digital-analog converters supplying the resolver tracks. • In the Lenze setting, the gains for both resolver tracks are preset identically.
Sine and cosine track do not magnetise orthogonally to each other.	Correction of the angle by means of which the two resolver tracks are supplied in a manner relative to one another.

The i700 servo inverter provides the possibility to determine the resolver error characteristic of the connected resolver which can be used for the automatic generation of optimal settings for minimising the resolver error (resolver error compensation).

Preconditions for the execution

- The motor is operated in speed open-loop control and servo control.
- During the resolver error identification, the speed should be constant, if possible, and greater than 500 rpm.

Response of the motor during the execution

- The response of the motor corresponds to the speed setpoint.
- The motor may be running rough during identification. This is due to the identification method.

How to identify the resolver error characteristic:

1. Set object 0x2822 (or 0x3022 for axis B) to "9".
   - The progress of the procedure is shown in object 0x2823 (or 0x3023 for axis B).
   - After successful completion of the resolver error identification, the following parameters of the resolver error compensation are automatically written. The resolver now operates with these settings.
2. For permanent storage: After the procedure has been completed, upload the changed parameters (0x2C44 or 0x3444 for axis B) from the i700 servo inverter into the Controller.

   The »EASY Starter« serves to save the parameter settings of the i700 servo inverter as parameter file (*.gdc). In the »PLC Designer«, this file can then be imported in the corresponding axis. ▶ Saving changed parameters safe against mains failure (📖 60)

Note!

The detected gain can take values between 0 ...100 %.
- With a setting of 0 %, the gain of the corresponding resolver track is only 95 % of the Lenze setting.
- With a sensible resolver error compensation only one of the two gains is adapted. The other remains at 100 %.

0x2C44 | 0x3444 - Resolver error compensation: Parameter

Sub.	Name	Lenze setting	Data type
▶ 1	Resolver error compensation: Angle	0	INTEGER_16
▶ 2	Resolver error compensation: Cosine track gain	100 %	UNSIGNED_16
▶ 3	Resolver error compensation: Sine track gain	100 %	UNSIGNED_16

Subindex 1: **Resolver error compensation: Angle**				
[´] = angular minutes				
Setting range (min. value | unit | max. value)			Lenze setting	
-100		100	0	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				INTEGER_16

Subindex 2: **Resolver error compensation: Cosine track gain**				
Setting range (min. value | unit | max. value)			Lenze setting	
0	%	100	100 %	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

Subindex 3: **Resolver error compensation: Sine track gain**				
Setting range (min. value | unit | max. value)			Lenze setting	
0	%	100	100 %	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

Deactivating the resolver error compensation

For the deactivation of the resolver error compensation, the corresponding parameters must be reset to the Lenze setting again.

5.9.3 Settings for "encoder" version

The use of an incremental sin/cos encoder is preset. If a sin/cos absolute value encoder with HIPERFACE® protocol is connected instead, select "2" in 0x2C40 (or 0x3440 for axis B) and adapt the encoder parameters (e.g. supply voltage) accordingly.

0x2C40 | 0x3440 - Encoder: Type

Selection list (Lenze setting printed in bold)			
1	**Sin/cos encoder**		
2	Hiperface absolute value encoder		
☑ Write access ☑ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_8

0x2C42 | 0x3442 - Encoder: Parameter

Sub.	Name	Lenze setting	Data type
1	Encoder: Increments / revolution	1024	UNSIGNED_32
2	Encoder: Supply voltage	5.0 V	UNSIGNED_8
3	Encoder: Angle drift - Actual angle error		INTEGER_16
4	Encoder: Signal quality - Actual amplitude		UNSIGNED_8

Subindex 1: **Encoder: Increments / revolution**				
Setting range (min. value | unit | max. value)			Lenze setting	
1		131072	1024	
☑ Write access ☑ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_32

Subindex 2: **Encoder: Supply voltage**				
Setting range (min. value | unit | max. value)			Lenze setting	
5.0	V	12.0	5.0 V	
☑ Write access ☑ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	UNSIGNED_8

Subindex 3: **Encoder: Angle drift - Actual angle error**				
From version 01.03				
Display area (min. value | unit | max. value)			Initialisation	
-3276.8	°	3276.7		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/10	INTEGER_16

Subindex 4: **Encoder: Signal quality - Actual amplitude**				
From version 01.03 The signal quality indicates the actual amplitude of the SinCos analog signals with regard to 1 Vss = 100 %. • In case of higher-order drives, the signal quality should be between 95 % and 105 %. • There is no need for optimisation if the signal quality is within the tolerance zone for the analog encoder signals given in the data sheet of the encoder manufacturer.				
Display area (min. value | unit | max. value)			Initialisation	
0	%	255		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☐ TX				UNSIGNED_8

Determination and display of the current angular drift

The value displayed in the 0x2C42:3 object (or 0x3422:3 for axis B) indicates whether too much or too less pulses, caused by EMC, have been detected by the device-internal counter unit. This value is detected differently, depending on the encoder type:

In case of an incremental sin/cos encoder, the pulses are counted between two zero pulse events of the Z track. In an error-free status, this value corresponds to the set number of increments. The accuracy of this procedure corresponds to a line graduation of the encoder of ± 1. The difference between set number of increments and counted pulses is converted to an angle with an accuracy of ±0.1°. A disadvantage is that only after one complete encoder revolution, an updated value of the angular drift is available and thus the update rate depends on the speed.

In case of a sin/cos absolute value encoder with HIPERFACE® protocol, no Z track is available. Here, instead, the position is regularly read out of the encoder. When the encoder is read out for the first time (after power-up or removal of an open circuit), the encoder position is used to initialise the device-internal counter unit and set a device-internal position. All other read-out processes from the encoder are used to generate a difference between the device-internal position and the encoder position. In an error-free status, the difference is zero. The accuracy of the process, however, is, speed-dependent due to the dead time of the communication with the encoder and thus restricted towards the zero pulse procedure. An advantage here, however, is that the update rate does not depend on the speed but on the communication rate only. The update rate is encoder-specific and generally is within the range between 30 ... 50 ms.

An evaluation of the angular drift regarding an error response is not provided in the Servo-Inverter i700. This has to be carried out in the control system.

5.9.4 Additional settings for SinCos absolute value encoders with HIPERFACE® protocol

Absolute value encoders are especially suitable for:

- Synchronous motors operated in the "servo control" mode. The synchronous motor (SM) servo control requires a pole position angle. This has to be detected only once during commissioning and saved as offset towards the absolute position in the axis data.
- Positioning modes in which homing is to be carried out only once.

The analog evaluation of the sin/cos tracks causes a high resolution. With regard to the storage of the position information, we distinguish between singleturn and multiturn encoders:

- Singleturn: Storage within one revolution
- Multiturn: Storage within a number of revolutions

Supported encoder types with HIPERFACE® protocol

The following encoder types are supported by the i700 servo inverter:

Type	Increments/revolution	Absolute revolutions	Type code (0x2C41:1 | 0x3441:1)
AM1024-8V-H (SRM50)	1024	4096 (Multiturn)	39
AM1024-8V-K2	1024		39
AM16-8V-H (SEL37)	16		71
AM16-8V-H (SEL52)	16		71
AM512-8V-H (SCM70)	512		7
AS1024-8V-H (SRS50)	1024	1 (Singleturn)	34
AS1024-8V-K2	1024		34
AS16-8V-H (SEK37)	16		66
AS16-8V-H (SEK52)	16		66
AS512-8V-H (SCS70)	512		2

Use of a non-supported encoder type

If an encoder is to be used, the type code of which is not listed in the table of the supported encoder types, this encoder can be introduced to the i700 servo inverter via the subindices 2 and 3 of the hiperface parameters described in the following. Please also observe the notes in the description of subindex 8.

Reading data out of the encoder

The "Determine data of the Hiperface encoder" function in the object 0x2822 (or 0x3022 for axis B) serves to read the type code, number of increments and number of distinguishable revolutions out of the encoder and automatically enter then into the corresponding Hiperface parameters.

0x2C41 | 0x3441 - Hiperface: Parameter

Sub.	Name	Lenze setting	Data type
1	Hiperface: Determined type code	0	UNSIGNED_8
2	Hiperface: User def. encoder - type code	0	UNSIGNED_8
3	Hiperface: User def. encoder - specifiable revolutions	1	UNSIGNED_16
4	Hiperface absolute value fault: Response	1: Trouble	UNSIGNED_8
5	Hiperface: Serial number		STRING(50)
6	Hiperface: Raw data - Actual position		UNSIGNED_32
7	Hiperface: Detected Increments / revolution		UNSIGNED_16
8	Hiperface: Type code supported by firmware	0: Not supported	UNSIGNED_8
9	Hiperface: Encoder type	0: Rotary transducer	UNSIGNED_8
10	Hiperface: Period length linear encoder		UNSIGNED_32

Subindex 1: **Hiperface: Determined type code**			
Type code read out of the encoder If a sin/cos encoder is set in 0x2C40 (or 0x3440 for axis B) or a communication error has occurred, this value is zero.			
Display area (min. value | unit | max. value)			Initialisation
0		255	0
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			UNSIGNED_8

Subindex 2: **Hiperface: User-defined encoder - type code**			
If the encoder is not supported by the firmware (see subindex 8): → Here, manually set the type code displayed in the subindex 1.			
Setting range (min. value | unit | max. value)			Lenze setting
0		255	0
☑ Write access ☑ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_8

Subindex 3: **Hiperface: User-defined encoder - number of revolutions**			
If the encoder is not supported by the firmware (see subindex 8): → Here, manually set the number of distinguishable revolutions. **Stop!** A wrong setting of the number of distinguishable revolutions may cause a breakdown!			
Setting range (min. value | unit | max. value)			Lenze setting
1		65535	1
☑ Write access ☑ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_16

Subindex 4: **Hiperface absolute value fault: Response**		
Selection of the response to communication problems or unknown encoder		
Selection list (Lenze setting printed in bold)		
0	No response	
1	**Fault**	
2	Warning	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX		UNSIGNED_8

Subindex 5: **Hiperface: Serial number**				
From version 01.03 The displayed serial number can be used for detecting an encoder exchange				
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX				STRING(50)

Subindex 6: **Hiperface: Raw data - Actual position**				
From version 01.03 The encoder-internal position value is output without being converted				
Display area (min. value | unit | max. value)			Initialisation	
0		4294967295		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX				UNSIGNED_32

Subindex 7: **Hiperface: Detected Increments / revolution**				
From version 01.03 Encoder increment according to type code or encoder nameplate				
Display area (min. value | unit | max. value)			Initialisation	
0		65535		
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☑ TX				UNSIGNED_16

Subindex 8: **Hiperface: Type code supported by firmware**			
From version 01.03 If an encoder is connected that is not supported by the firmware, it will be displayed here. In this case, the same response takes place as in case of a communication error. The error can be removed by manually setting the type code displayed in subindex 1 in subindex 2. This serves to signalise to the firmware that the number of distinguishable revolutions is as well set correctly in the subindex 3 by the user.			
Selection list (read only)			
0	**Not supported**		
1	Supported by firmware		
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☑ TX			UNSIGNED_8

Subindex 9: **Hiperface: Encoder type**			
From version 01.05 Detected encoder type (rotary/linear)			
Selection list (read only)			
0	**Rotary encoder**		
1	Linear encoder		
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☑ TX			UNSIGNED_8

Subindex 10: **Hiperface: Period length of linear encoders**				
From version 01.05 In case of a linear encoder, here the period length or scaling is displayed in [nm]. In case of a rotary encoder, the value "0" is displayed.				
Display area (min. value | unit | max. value)			Initialisation	
0	nm	4294967295		
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☑ TX				UNSIGNED_32

Communication error monitoring

Communication with the encoder is monitored by the protocol and by generating a checksum. If a violation of the communication protocol or a defect frame is detected, a response occurs as a function of the device status and the setting in subindex 4 (Hiperface communication error: Response):

Device status	**Response in the event of an error** (depending on the error responses set in the Subindex 4)		
	0: No Response	1: Trouble	2: Warning
Not ready to switch on	Warning		
Switch on disabled	Warning		
Ready to switch on	-	Fault	Warning
Switched on	-	Fault	Warning
Operation enabled	-	Fault	Warning
Quick stop is active	-	Fault	Warning
Fault reaction active	-	Warning	
Fault	-	Warning	

5.9.5 Detection of changed settings of the feedback system

Bit 0 of the Lenze statusword 2 (0x2833 or 0x3033 for axis B) indicates whether the settings of the feedback system have been changed during operation. Bit 0 is set to"1" if one of the following objects have been changed since the last controller inhibit:

Object		**Name**
Axis A	Axis B	
0x2C40	0x3440	Encoder: Type
0x2C41:2	0x3441:2	Hiperface: User defined - Type code
0x2C41:3	0x3441:3	Hiperface: User defined - Number of revolutions
0x2C41:5	0x3441:5	Hiperface: Serial number
0x2C42:1	0x3442:1	Encoder: Increments / revolution
0x608F	0x688F	Position encoder resolution

After a controller enable, bit 0 of the Lenze status word 2 is always reset to "0".

5.10 Synchronous motor (SM): Pole position identification

Note!

Only required:
- For servo control with synchronous motor of a third-party manufacturer.
- For servo control with synchronous motor and use of incremental encoders (TTL or sin/cos encoders as well as resolvers).
- After changes of the motor feedback system, e.g. encoder exchange.

For the control of a permanent-magnet synchronous motor, the pole position – the angle between motor phase U and the field axis of the rotor – must be known.

- For Lenze motors with an absolute value encoder or resolver, the pole position is already set correctly.
- When incremental encoders (TTL or sin/cos encoders without absolute position information) are used, a pole position identification (PPI) is always required after switching on the mains (initialisation), even with Lenze motors.

Functions for pole position identification

For the identification of the pole position of the currently activated motor encoder, the following three functions are available:

- Pole position identification PPI (360°)
- Pole position identification PPI (min. movement)
- Pole position identification PPI (without movement)

Danger!

In case of firmly braked motors (e.g. for hanging loads), only the "pole position identification PLI (without motion)" function may be used!

For the other two functions "Pole position identification PLI (360°)" and "Pole position identification PLI (min. motion)" the motor must not be braked or blocked during the pole position identification!

The functions should yield similar results. Due to e.g. friction, bearing forces, and a trapezoidal field profile, the results may deviate. Here, the function that includes one full revolution (360°) will provide the most precise results and the function without any motion will provide the most inaccurate results. An increase in the percentage of the current amplitude helps provide more precise results.

You can find detailed information on the respective function in the following subchapters.

5.10.1 Monitoring of the pole position identification

If an error occurs during the pole position identification or the pulse inhibit gets active (e.g. due to short-time undervoltage), the procedure is terminated with controller inhibit without a change in settings.

If the motor was braked or blocked during the procedure, this will be recognised at the end of the measurement and no change is made either (exception: "Pole position identification PLI (without motion").

The error response can be parameterised:

0x2C60 | 0x3460 - Monitoring of pole position identification: Response

If the pole position identification is aborted, the response set here is triggered.
- If this behaviour is not wanted, deactivate the monitoring by selecting "0: No response".

Selection list (Lenze setting printed in bold)		
0	No response	
1	**Fault**	
2	Warning	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX		UNSIGNED_8

5.10.2 Pole position identification PPI (360°)

Danger!

The motor must not be braked or blocked during the pole position identification! For this reason, this function is not permitted for hanging loads!

Stop!

Check the correct parameterisation of the followng monitoring modes before carrying out the pole position identification to prevent the motor from being permanently damaged in the event of an error:
- Monitoring of the motor utilisation (I^2xt) (📖 241)
- Monitoring of the ultimate motor current (📖 258)

Functional description

If servo control for synchronous motor is set and if no error is pending, the current is increased to 141 % of the rated motor current along a ramp first after controller enable. The rotor thus provides the necessary basis for further proceeding.

The commutation angle is changed (for 2 s) from 45° to 0° along a ramp. This brings the rotor reliably into a defined position. Afterwards, the commutation angle is changed from 0° to 360° along a ramp, passing through approximately 45° every 4 s. Thus, the motor carries out a complete electrical revolution, resulting in a mechanical rotation of the motor shaft in positive direction (clockwise when looking at the A-side of the motor). Within the 360° several measurements are carried out and evaluated.

Preconditions for the execution

- The motor must not be braked or blocked during the pole position identification.
- The controller is free of errors and is in the "Switched on" device status.

Response of the motor during the execution

During the pole position identification the rotor aligns itself. The motor shaft moves by max. one electrical revolution which causes the corresponding movement of the connected mechanical components!

How to execute the pole position identification PPI (360°):

1. If the controller is enabled, inhibit the controller.
   ▸ Enable/inhibit operation via control word (📖 59)
2. Set object 0x2825 (or 0x3025 for axis B) to "5" to change to the "Pole position identification PPI (360°)" operating mode.
3. Enable the controller to start the procedure.

   Note: By means of controller inhibit, the procedure started can be cancelled any time, if required, without a change in settings.

After successful completion of the pole position identification...

...controller inhibit is set automatically and the pole position specified in object 0x2C03:2 (or 0x3403:2 for axis B) for the activated feedback system is set.

- For permanent storage, the changed settings must be uploaded to the controller from the i700 servo inverter.
  The »EASY Starter« serves to save the parameter settings of the i700 servo inverter as parameter file (*.gdc). In the »PLC Designer«, this file can then be imported in the corresponding axis.
  ▸ Saving changed parameters safe against mains failure (📖 60)
- The controller inhibit automatically set by the procedure can be deactivated via the Controlword (0x6040 or 0x6840 for axis B). ▸ Enable/inhibit operation via control word (📖 59)

Adapt pole position identification PPI (360°)

The above-described procedure for pole position identification can be adjusted to the respective machine and the prevailing moments of inertia by means of the parameters described in the following.

Stop!

If there is no temperature monitoring in the motor and/or the I²xt motor monitoring and the maximum current monitoring are not parameterised correctly, the motor might be damaged permanently when the current amplitude is set too high (e.g. to the maximum value!

▸ Monitoring of the motor utilisation (I²xt) (📖 241)

▸ Monitoring of the ultimate motor current (📖 258)

0x2C61 | 0x3461 - Pole position identification PPI (360°)

Sub.	Name	Lenze setting	Data type
1	PPI (360°): Current amplitude	100 %	UNSIGNED_16
2	PPI (360°): Ramp time	40 s	UNSIGNED_16
3	PPI (360°): Direction of rotation	0: Field: CW	UNSIGNED_8
4	PPI (360°): Error tolerance	20 °	UNSIGNED_8
5	PPI (360°): Absolute current amplitude		UNSIGNED_32

Subindex 1: **PPI (360°): Current amplitude**			
Adjustment of the current amplitude in percent • For large machines and high mass inertia values or for linear direct drives, the current amplitude usually has to be increased. • A Lenze setting of "100 %" corresponds to 141 % of the rated motor current (0x6075 or 0x6875 for axis B). **Note!** If the current amplitude is set to > 100 %, the device utilisation (Ixt) monitoring and/or one of the motor monitoring functions may respond and cause the abort of the pole position identification.			
Setting range (min. value | unit | max. value)			Lenze setting
1	%	1000	100 %
☑ Write access ☑ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_16

Subindex 2: **PPI (360°): Ramp time**			
Adjustment of the ramp time in percent • For large machines and high mass inertia values, the ramp time usually has to be increased. • For small machines, a reduction of the ramp time can speed up the pole position identification process.			
Setting range (min. value | unit | max. value)			Lenze setting
1	s	600	40 s
☑ Write access ☑ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_16

Subindex 3: **PPI (360°): Direction of rotation**		
Selection of the travel direction • In some situations it may be helpful to reverse the travel direction for the pole position identification (e.g. for linear motor at the end stop).		
Selection list (Lenze setting printed in bold)		
0	**Field: CW**	
1	Field: CCW	
☑ Write access ☑ CINH ☐ OSC ☑ P ☐ RX ☐ TX		UNSIGNED_8

Subindex 4: **PPI (360°): Error tolerance**			
Error tolerance for the plausibility check • If the rotor position detected via the encoder system is not within the tolerance zone around the position that is output in a controlled manner, the pole position identification is aborted and the error response parameterised is tripped. • When optimising the pole position identification by means of the objects listed before, it may be sensible to first increase the tolerance zone in order to evaluate the entire throughput based on an oscillscope recording. This is especially required in case of mechanically coupled motors, as here, the preset tolerance zone is quickly exceeded due to the feedbacks of mass inertia, friction and static friction. After the deviations with iteratively optimised PPI behaviour are in the common tolerance zone again, the value can/has to be reduced to 20 ° again.			
Setting range (min. value | unit | max. value)			Lenze setting
15	°	50	20 °
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_8

Subindex 5: **PPI (360°): Absolute current amplitude**				
Display of the absolute current amplitude				
Display area (min. value | unit | max. value)			Initialisation	
0.00	A	42949672.95		
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			Scaling: 1/100	UNSIGNED_32

5.10.3 Pole position identification PPI (min. movement)

Danger!

The motor must not be braked or blocked during the pole position identification! For this reason, this function is not permitted for hanging loads!

Stop!

Check the correct parameterisation of the followng monitoring modes before carrying out the pole position identification to prevent the motor from being permanently damaged in the event of an error:

- Monitoring of the motor utilisation (I²xt) (📖 241)
- Monitoring of the ultimate motor current (📖 258)

Functional description

If servo control for synchronous motor is set and if no error is pending, the current position is memorised after controller enable, and the current is increased along a ramp for 10 s to 35 % of the rated motor current. This will cause the rotor to align, which, however, is compensated by a position control. If the rotor makes an electrical movement of more than 20°, an error message is output, and the value measured is discarded. This might occur in the case of motors with considerable detent torques.

If the current has reached its final value, a plausibility check is executed after a short interval: In order to detect a non-permissible blocking of the motor, a positive and negative test angle (± 20°) relative to the current position are defined after the identification. The motor must align itself to these two test angles within a tolerance of 25 %.

Preconditions for the execution

- The motor must not be braked or blocked during the pole position identification.
- The controller is free of errors and is in the "Switched on" device status.

Response of the motor during the execution

The motion of the motor will maximally correspond to the set "Max. permissible motion" (Lenze setting: 20°). If a greater motion is detected via the encoder system, the pole position identification is cancelled and the parameterised error response (Lenze setting: Fault) is triggered.

How to execute the pole position identification PPI (min. movement):

1. If the controller is enabled, inhibit the controller.
   ▸ Enable/inhibit operation via control word (📖 59)
2. Set object 0x2825 (or 0x3025 for axis B) to "6" to change to the "Pole position identification (min. movement)" operating mode.
3. Enable the controller to start the procedure.

   Note: By means of controller inhibit, the procedure started can be cancelled any time, if required, without a change in settings.

After successful completion of the pole position identification...

...controller inhibit is set automatically and the pole position specified in object 0x2C03:2 (or 0x3403:2 for axis B) for the activated feedback system is set.

- For permanent storage, the changed settings must be uploaded to the controller from the i700 servo inverter.
  The »EASY Starter« serves to save the parameter settings of the i700 servo inverter as parameter file (*.gdc). In the »PLC Designer«, this file can then be imported in the corresponding axis.
  ▸ Saving changed parameters safe against mains failure (📖 60)
- The controller inhibit automatically set by the procedure can be deactivated via the Controlword (0x6040 or 0x6840 for axis B). ▸ Enable/inhibit operation via control word (📖 59)

Adapt pole position identification PPI (min. movement)

The above-described procedure for pole position identification can be adjusted to the respective machine and the prevailing moments of inertia by means of the parameters described in the following.

Stop!

If there is no temperature monitoring in the motor and/or the I²xt motor monitoring and the maximum current monitoring are not parameterised correctly, the motor might be damaged permanently when the current amplitude is set too high (e.g. to the maximum value!

▸ Monitoring of the motor utilisation (I²xt) (📖 241)
▸ Monitoring of the ultimate motor current (📖 258)

0x2C62 | 0x3462 - Pole position identification PPI (min. movement)

Sub.	Name	Lenze setting	Data type
1	PPI (min. movement): Current amplitude	25 %	UNSIGNED_16
2	PPI (min. movement): Ramp time - current	10 s	UNSIGNED_16
3	PPI (min. movement): Gain	0 %	UNSIGNED_16
4	PPI (min. movement): Reset time	62.5 ms	UNSIGNED_16
5	PPI (min. movement): Max. move permitted	20 °	UNSIGNED_8
6	PPI (min. movement): Absolute current amplitude		UNSIGNED_32

Subindex 1: **PPI (min. movement): Current amplitude**			
Adjustment of the current amplitude in percent • For large machines and high mass inertia values or for linear direct drives, the current amplitude usually has to be increased. • A Lenze setting of "100 %" corresponds to 35 % of the rated motor current (0x6075 or 0x6875 for axis B). **Note!** If the current amplitude is set to > 400 %, the device utilisation (Ixt) monitoring and/or one of the motor monitoring functions may respond and cause the abort of the pole position identification.			
Setting range (min. value | unit | max. value)			Lenze setting
1	%	1000	25 %
☑ Write access ☑ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_16

Subindex 2: **PPI (min. movement): Ramp time - current**			
Adjustment of the rate of rise of the current in percent			
Setting range (min. value | unit | max. value)			Lenze setting
1	s	600	10 s
☑ Write access ☑ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_16

Subindex 3: **PPI (min. movement): Gain**			
Adjustment of the proportional gain of the PI controller • With the Lenze setting "0", the PI controller works as an I-controller.			
Setting range (min. value | unit | max. value)			Lenze setting
0	%	1000	0 %
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_16

Subindex 4: **PPI (min. movement): Reset time**			

Adjustment of the PI controller reset time
• For this, observe the following setting details:
  • In order to be able to compensate a positional variation faster, first the reset time should be reduced. If this does not result in the desired behaviour, the proportional gain can be increased.
  • Ensure that the position control does not get unstable. We therefore recommend to use an I controller.			

Setting range (min. value | unit | max. value)			Lenze setting
0.1	ms	6000.0	62.5 ms
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10 UNSIGNED_16

Subindex 5: **PPI (min. movement): Max. move permitted**				
Adjustment of the permitted movement • The pole position identification comprises a monitoring function for the follow-up control. If a movement greater than the permissible movement set is detected by the encoder system, the pole position identification is aborted and the error response parameterised is tripped: • In order to detect a non-permissible blocking of the machine, a positive and negative test angle relative to the current position are defined after the identification. The machine must align itself to these two test angles within a tolerance of 25 %. The size of the test angle corresponds to the max. move permitted set here.				
Setting range (min. value | unit | max. value)			Lenze setting	
1	°	90	20 °	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_8

Subindex 6: **PPI (min. movement): Absolute current amplitude**				
Display of the absolute current amplitude				
Display area (min. value | unit | max. value)			Initialisation	
0.00	A	42949672.95		
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			Scaling: 1/100	UNSIGNED_32

5.10.4 Pole position identification PPI (without movement)

This function extension is available from version 01.03!

Note!

Due to a slight inaccuracy in this PLI process, higher variations regarding the detected pole position have to be expected. This also causes different torque efficiencies. Thus, we recommend the use of higher-order PLI processes ("360 °" or "min. movement"). If this is not possible for process reasons, we recommend the execution of the pole position identification PLI (without movement) only after the controller is enabled for the first time. ▸ Optional settings (starting performance)

During the pole position identification, the error 0xFF13 ("identification cancelled") may occur. This may be an indication that the motor features are not suitable for this PLI process.

Functional description

After controller enable, a defined pulse pattern is output which provides currents up to approximately maximum motor current. The respective currents are measured. Based on these currents, the field distribution can be detected so that the pole position can be calculated. Afterwards, the controller is inhibited automatically.

The pole position identification (without movement) does not require any further parameterisation.

Preconditions for the execution

- The motor may be firmly braked.
- The controller is free of errors and is in the "Switched on" device status.

Response of the motor during the execution

Only a few short "knocks" can be heard from the motor, caused by the current test pulses.

How to execute the pole position identification (without movement):

1. If the controller is enabled, inhibit the controller.
   ▶ Enable/inhibit operation via control word (📖 59)
2. Set object 0x2825 (or 0x3025 for axis B) to "7" to change to the "Pole position identification PPI (without movement)" operating mode.
3. Enable the controller to start the procedure.

   Note: By means of controller inhibit, the procedure started can be cancelled any time, if required, without a change in settings.

After successful completion of the pole position identification...

...controller inhibit is set automatically and the pole position specified in object 0x2C03:2 (or 0x3403:2 for axis B) for the activated feedback system is set.

- For permanent storage, the changed settings must be uploaded to the controller from the i700 servo inverter.

  The »EASY Starter« serves to save the parameter settings of the i700 servo inverter as parameter file (*.gdc). In the »PLC Designer«, this file can then be imported in the corresponding axis.
  ▶ Saving changed parameters safe against mains failure (📖 60)
- The controller inhibit automatically set by the procedure can be deactivated via the Controlword (0x6040 or 0x6840 for axis B). ▶ Enable/inhibit operation via control word (📖 59)

Optional settings (starting performance)

Optionally, a PLI without motion can be activated after switching on the i700 servo inverter.

Danger!

Each pole position identification causes an update of the pole position set in the device! Therefore, ensure that the response to open circuit in the feedback system is set to Lenze setting "1: Fault" in 0x2C45 (or 0x3445 for axis B)!
Otherwise, the status of the feedback system in case of open circuit is undefined and the pole position can assume any values. There is a danger that the machine accelerates in an uncontrolled way after pole position identification!

Note!

The process of the pole position identification only lasts some milliseconds. During the pole position identification, the device status does not change. Only after the pole position identification, the "Switched on" device status changes to the "Operation enabled" device status.

0x2C63 | 0x3463 - pole position identification PLI (without movement)

From version 01.05

Sub.	Name	Lenze setting	Data type
▸ 1	Behaviour after switch-on	0: No operation	UNSIGNED_8

Subindex 1: **Behaviour after switch-on**			
Starting performance (without or with PLI before starting) in case of synchronous motor (SM) servo control • In case of all other control modes, the setting has no influence.			
Selection list (Lenze setting printed in bold)		Info	
0	**No Operation**	In the Lenze setting, the behaviour is the same as before.	
1	PPI at initial switch-on and after encoder error only	After the first controller enable and after each encoder wire breakage, a PLI without movement takes place. **Note:** In order that the PLI is executed after each encoder open circuit, the response to open circuit in the feedback system has to be set to the Lenze setting "1: Fault" in 0x2C45 (or 0x3445 for axis B). Otherwise, the status of the feedback system in case of open circuit is undefined and the pole position can assume any values. There is a danger that the machine accelerates in an uncontrolled way after pole position identification!	
2	PPI after every switch-on	After every controller enable, a PLI without movement is carried out in advance.	
☑ Write access ☑ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_8

5.11 Setting control loops

Subsequent to the motor parameterisation, the different control loops must be set. For quick commissioning, the calculations and settings can be carried out automatically using the commissioning functions of the servo inverter. For manual setting, applicable equations are offered in the following subchapters, too.

The following table shows the corresponding commissioning steps required for the different control types and motors:

Commissioning step	Servo control SM	Servo control ASM	V/f characteristic control
Setting and optimising the current controller	(●)	(●)	●[1]
Determining the total moment of inertia	●	●	
Setting the speed controller	●	●	
Setting the position controller	●	●	
Setting the field controller (ASM)		(●)	
Setting the field weakening controller (ASM)		(●)	
Information on the following control loops/commissioning steps for the V/f characteristic control can be found in the chapter "Parameterising the V/f characteristic control"			
Set Imin controller ▶ Activating the voltage vector control (Imin controller)			●
Set Imax controller ▶ Defining the behaviour at the current limit (Imax controller)			●
Set "restart on the fly" controller ▶ "Flying restart" function			●

● Required

(●) Only required for other manufacturers' motors

[1] Only required if voltage vector control, DC-injection braking, or flying restart process is activated.

5.11.1 Setting and optimising the current controller

Note!

For a servo control, the current controller should always be optimised if a motor of another manufacturer with unknown motor data is used!

For a V/f characteristic control, the current controller only has to be optimised if voltage vector control is used, or if DC-injection braking or the flying restart process is activated.

The control system includes two current controllers, a direct-axis current controller and a cross current controller, whose parameterisation is carried out identically. The direct-axis current controller controls the field-producing current (D current). The cross current controller controls the torque-producing current (Q current). There is a coupling between the two control loops which makes every actuation of every one of the controllers occur as fault in the control loop of the other controller. This coupling can be compensated for by activating the current controller feedforward control via object 0x2941 (or 0x3141 for axis B).

For the automatic calculation of the two controller parameters, the "Current controller: Calculate controller parameters" function is provided via object 0x2822 (or 0x3022 for axis B).

- The calculating function is based on the stator resistance (0x2C01:2 or 0x3401:2 for axis B) and the stator leakage inductance (0x2C01:3 or 0x3401:3 for axis B). Thus, these motor parameters have to be parameterised before. This can be done either by entering the data sheet values manually or by a motor parameter identification run.

The calculated controller parameters can be optimised by means of an experimental adjustment subsequently. The procedure is described in the following subchapter, Manual test mode "Current pulse".

Equations for calculating the gain and the reset time for the synchronous motor

$$V_p = \frac{L_{ss}}{T_{Dead\ time}} \qquad T_n = \frac{L_{ss}}{R_s}$$

Axis A	Axis B	Symbol	Description	Dimension unit
0x2942:1	0x3142:1	V_p	Current controller gain	V/A
0x2C01:3	0x3401:3	L_{ss}	Motor stator inductance	H
-	-	$T_{dead\ time}$	Equivalent time constant for analog detection and scanning = 0.00034 s (340 µs)	s
0x2942:2	0x3142:2	T_n	Current contr. reset time	s
0x2C01:2	0x3401:2	R_s	Motor stator resistance (value at 20°C)	Ω

Equations for calculating the gain and the reset time for the asynchronous motor

$$V_p = \frac{\sigma \cdot L_s}{T_{Dead\ time}} \approx \frac{2 \cdot L_{ss}}{T_{Dead\ time}} \qquad T_n = \frac{\sigma \cdot L_s}{R_s} \approx \frac{2 \cdot L_{ss}}{R_s}$$

Axis A	Axis B	Symbol	Description	Dimension unit
0x2942:1	0x3142:1	V_p	Current controller gain	V/A
-	-	σ	Leakage	
-	-	L_s	Motor stator inductance	H
0x2C01:3	0x3401:3	L_{ss}	Motor stator leakage inductance	H
-	-	$T_{dead\ time}$	Equivalent time constant for analog detection and scanning = 0.00034 s (340 µs)	s
0x2942:2	0x3142:2	T_n	Current contr. reset time	s
0x2C01:2	0x3401:2	R_s	Motor stator resistance (value at 20°C)	Ω

Tip!

To ensure a smooth transition to the field weakening range, it is recommended to apply the equations for the synchronous motor for the asynchronous motor as well.

0x2941 | 0x3141 - Current controller: Feedforward control

Activate/deactivate the current controller feedforward control
- Since the positioning movements are known, they can be precontrolled in order to slightly increase the achievable dynamic performance of the control loop.

Note!
A successful feedforward control requires knowledge of the equivalent circuit data of the motor. If only estimated values are available, the feedforward control should not be activated.

Selection list (Lenze setting printed in bold)			
0	**Off**		
1	Activate		
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_8

0x2942 | 0x3142 - Current controller: Parameter

Sub.	Name	Lenze setting	Data type
1	Current controller: Gain	148.21 V/A	UNSIGNED_32
2	Current controller: Reset time	3.77 ms	UNSIGNED_32

Subindex 1: **Current controller: Gain**				
Setting range (min. value | unit | max. value)			Lenze setting	
0.00	V/A	750.00	148.21 V/A	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/100	UNSIGNED_32

Subindex 2: **Current controller: Reset time**				
Setting range (min. value | unit | max. value)			Lenze setting	
0.01	ms	2000.00	3.77 ms	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/100	UNSIGNED_32

0x2943 | 0x3143 - Motor: Current setpoint - filter time

Setting range (min. value | unit | max. value)			Lenze setting	
0.00	ms	10.00	0.00 ms	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/100	UNSIGNED_16

5.11.1.1 Manual test mode "Current pulse"

The current controller must be adapted to the electrical characteristics of the motor – stator resistance and stator inductance. For an experimental adjustment, the manual test mode "Current pulse" can be used.

Note!

This test mode is intended for the adjustment of the current controller in the "servo control for synchronous motor/asynchronous motor" mode and not for the adjustment of the Imax controller in the "V/f characteristic control (VFC)" mode!

Functional description

In the manual "Current pulse" test mode, setpoint step-changes are applied to the current controller input after controller enable. The step responses then either have to be recorded using the oscilloscope and clamp-on ammeter, or by means of the i700 servo inverter oscilloscope function. By evaluating the step responses, it is the objective to optimise the two current controller parameters "gain" and "reset time" so that a quick current characteristic free of harmonics is obtained. (See the following instructions.)

Motors with individual pole windings, which can feature very distinct saturation phenomena in the stator leakage inductance, satisfactory results are possibly only achieved with a current-dependent correction of the current controller parameters. For this purpose, a characteristic is stored in the i700 servo inverter, describing the current dependence of the stator leakage inductance and correcting the current controller gain. ▸ Correction of the stator leakage inductance (Lss)...

Stop!

In the case of the synchronous motor, a jerky compensating movement occurs after controller enable if the pole position of this movement does not correspond to the starting angle.

Motor phase U is supplied with a DC current, the level of which is determined via the following equation on the left. Motor phases V and W then carry half of this DC current, respectively (negative; from the motor).

Equations for calculating the DC currents in motor phases U, V, W

$$I_{phase\_U} = \sqrt{2} \cdot I_{test}[\%] \cdot \frac{I_{rated}}{100\,\%} \qquad I_{phase\_V,\_W} = -0.5 \cdot \sqrt{2} \cdot I_{test}[\%] \cdot \frac{I_{rated}}{100\,\%}$$

Axis A	Axis B	Symbol	Description	Dimension unit
0x2D83:2	0x3583:2	I_{phase_U}	Present current in motor phase U	A
0x2D83:3	0x3583:3	I_{phase_V}	Present current in motor phase V	A
0x2D83:4	0x3583:4	I_{phase_W}	Present current in motor phase W	A
0x2835:1	0x3035:1	I_{test}	Setpoint current for manual test mode	%
0x6075	0x6875	I_{rated}	Rated motor current	A

Greyed out = read access only

Preconditions for the execution

- The motor has to be completely parameterised.
- The monitoring of the motor utilisation (I²xt) has to be parameterised and switched actively.
  ▸ Monitoring of the motor utilisation (I²xt) (📖 241)
- The motor must be able to rotate freely.
- The controller is free of errors and is in the "Switched on" device status.
- For the test, the rotor of synchronous motors has to be positioned in the pole centre. Some synchronous motors may require an alignment and blocking in the pole centre.
  - The use of the manual test mode "current/frequency" for one-time alignment of the rotor is reasonable with the following settings:
    R.m.s. value = 70 ... 100 %; frequency = 0 Hz; Starting angle = 0°
    ▸ Manual test mode "current/frequency" (📖 63)
  - Afterwards fixing by applying the holding brake or using external fixing tools.

Stop!

After the current controller has been adjusted, remove the mechanical fixing again!

Response of the motor during the execution

The motor will align during the first controller enable, then usually it won't align anymore.

How to adjust the current controller by means of the manual test mode "current pulse":

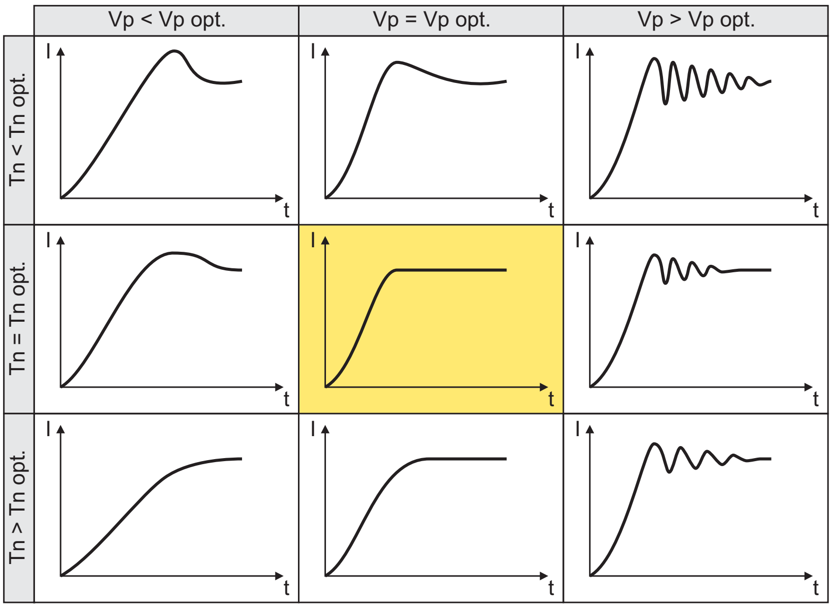
1. If the controller is enabled, inhibit the controller.
   ▸ Enable/inhibit operation via control word (📖 59)
2. Calculate the starting parameters for the current controller based on the parameterised motor data.
   - For the automatic calculation, the "Current controller: Calculate controller parameters" function is available via the object 0x2822 (or 0x3022 for axis B). ▸ Setting and optimising the current controller (📖 105)
   - Optionally, a manual determination is possible as well.
3. Set object 0x2825 (or 0x3025 for axis B) to "3" to change to the "Current pulse" test mode.
4. Set setpoint current for manual test mode in the object 0x2835:1 (or 0x3035:1 for axis B).

   The number of test points for the setpoint current results from the permissible current range of the motor and its characteristic (e.g. the Lss saturation for PM servo motors).
5. Enable the controller temporarily to start the test mode.

   After 100 ms, controller enable is cancelled automatically by the controller. Thus the danger of an undesired motor heat-up is avoided.

6. Record the step response of the motor current with the oscilloscope function of the »PLC Designer«/»EASY Starter«.
   - Parameters to be recorded:
     Actual D current (0x2DD1:1 or 0x35D1:1 for axis B)
     Setpoint D current (0x2DD1:3 or 0x35D1:3 for axis B)
     D voltage (0x2DD7:4 or 0x35D7:4 for axis B)
   - Oscilloscope settings: Sampling rate = 0.0625 ms; time base = 2 ... 5 ms

   Alternatively, the step response of the motor current in the motor phase U can be measured by means of an oscilloscope and clamp-on ammeter.
7. Evaluate the step response:



8. Adjust the gain and reset time of the current controller.
9. Repeat steps 3 ... 6 iteratively until the optimum step response of the motor current is reached.
   - In the optimised state the current rise time typically is 0.5 ... 1 ms.
   - If the adjustment does not provide any satisfactory results, the current controller feedforward control via object 0x2941 (or 0x3141 for axis B) can be activated additionally. Then steps 3 ... 6 are to be repeated.
10. To stop the test mode:
    - Inhibit controller.
    - Set object 0x2825 (or 0x3025 for axis B) to "0" to change back to the CiA402 mode.
11. For permanent storage: Upload changed current controller parameters (0x2942 or 0x3142 for axis B) from the i700 servo inverter into the Controller.

    The »EASY Starter« serves to save the parameter settings of the i700 servo inverter as parameter file (*.gdc). In the »PLC Designer«, this file can then be imported in the corresponding axis. ▸ Saving changed parameters safe against mains failure (📖 60)

5.11.2 Determining the total moment of inertia

The total moment of inertia is composed as follows:

Equation for calculating the total moment of inertia				
$J_{Total} = J_{Motor} + J_{Holding\ brake} + J_{Gearbox}^{*} + J_{Load}^{*}$				
Axis A	Axis B	Symbol	Description	Dimension unit
-	-	J_{total}	Total moment of inertia	$kg\ m^2$
0x2910:1	0x3110:1	J_{Motor}	When a Lenze motor is selected, this value is automatically set.	$kg\ cm^2$
-	-	$J_{Holding\ brake}$	This value can be determined from the catalog data and added to the values mentioned above.	$kg\ m^2$
0x2910:2	0x3110:2	$J_{Gearbox}^{*}$ J_{Load}^{*}	Sum of all other moments of inertia of the system. Here, it is to be considered that these are values that have to be transformed to the motor shaft (marked by *).	$kg\ cm^2$

For determining the total moment of inertia, the i700 servo inverter is triggered by the higher-level Controller and a test path (motion profile) is executed. After a mathematical procedure, the total mass moment of inertia is determined from the detected speed and torque characteristics. The load moment of inertia results from the difference between the total moment of inertia and the moment of inertia of the motor (techn. data of the motor).

For determining the other moments of inertia ($J_{Gearbox} + J_{Load}$), there are the following options:

- Extraction from the rated data of the drive dimensioning
- Determination based on data sheets of the components
- Calculation/estimation (for simple kinematic arrangements)
- Empirical determination by test mode with trapezoidal motion profile

For the calculation, you can use e.g. the new Lenze app "Formulae and tables".

Tip!

The "Formulae and tables" app for smartphones can be found in the internet:

http://www.lenze.com → Download

The "Formulae and tables" app contains basic formulae of kinematics, dynamics and force and torques as well as work, power, energy and control engineering and can be filled with data at any time.

The sum of the set moments of inertia ($J_{Motor} + J_{Gearbox} + J_{Load}$) forms the basis for an initial calculation of the speed controller gain with the "Speed controller: Calculate controller parameters" (0x2822 or 0x3022 = 5).

Note!

The reduced moment of inertia of the load (0x2910:2 or 0x3110:2 for axis B) is included
- directly in the calculation of the speed controller (0x2822 or 0x3022 = 5) and
- indirectly in the calculation of the position controller (0x2822 or 0x3022 = 6).

Thus, give the data with all due consideration.

0x2910 | 0x3110 - Moments of inertia

Sub.	Name	Lenze setting	Data type
1	Moment of inertia: Motor	0.14 kg cm²	UNSIGNED_32
2	Moment of inertia: Load	0.00 kg cm²	UNSIGNED_32
3	Moment of inertia: Motor-load coupling	0: Rigid system	UNSIGNED_8

Subindex 1: **Moment of inertia: Motor**				
Setting range (min. value | unit | max. value)			Lenze setting	
0.00	kg cm²	20000000.00	0.14 kg cm²	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/100	UNSIGNED_32

Subindex 2: **Moment of inertia: Load**				
Setting range (min. value | unit | max. value)			Lenze setting	
0.00	kg cm²	20000000.00	0.00 kg cm²	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/100	UNSIGNED_32

Subindex 3: **Moment of inertia: Motor-load coupling**			
Selection list (Lenze setting printed in bold)			
0	**Rigid system**		
1	Elastic system		
2	System with backlash		
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_8

5.11.3 Setting the speed controller

For the automatic calculation of the gain and the reset time, the "Speed controller: Calculate controller parameters" function is provided via object 0x2822 (or 0x3022 for axis B). This serves to calculate speed controller starting values. To ensure optimal operation, we recommend a manual follow-up optimisation.

The following equations apply to a rigid system. For elastic systems and systems with backlash, the gain is reduced. The moment of inertia required for the calculation usually consists of the moment of inertia of the motor and the load moments of inertia transformed to the motor side.

Equations for calculating the gain and the reset time (applicable for rigid systems)				
$V_p = \frac{J}{a \cdot (T_{Filter} + T_{Current\ controller})} \cdot \frac{2\pi}{60}$			$T_n = a^2 \cdot (T_{Filter} + T_{Current\ controller})$	
Axis A	Axis B	Symbol	Description	Dimension unit
0x2900:1	0x3100:1	V_p	Speed controller gain	Nm/rpm
-	-	J	Moment of inertia = J_{Motor} + sum (J_{Load})	kg m^2
-	-	A	Unit for the phase reserve (recommendation: a = 4 ≡ phase reserve of 60°)	
0x2904	0x3104	T_{filter}	Actual speed value filter time constant	s
-	-	$T_{current\ controller}$	Equivalent time constant of the current control loop = 0.0005 s (500 µs)	s
0x2900:2	0x3100:2	T_n	Speed controller reset time	s

Special case of the linear motor

Here a re-calculation from a linear system to a rotary system must be made. Therefore, via the feedback system a degree of freedom results for the determination of the number of pole pairs.

- For a rotary system, the number of pole pairs specifies the ratio of electrical and mechanical revolution, the number of encoder increments being defined via one mechanical revolution.
- In the case of a linear system, the user is free to decide for which length he or she wants to specify the number of encoder increments. Usually, the number of increments is given for a pole distance or for the total length of the linear scale. If the number of increments = "number of increments for one pole distance" is selected, a motor with the number of pole pairs zp = 1 is created.

The effective moment of inertia for a linear motor can be calculated according to the following equations. With this J value, the equations shown above can be used to calculate the speed controller gain and reset time.

Equations for calculating the effective moment of inertia for the linear motor				
$zp = \text{Integer}\left(\frac{s}{2\tau_{Pole\ pair}}\right)$			$J = m \cdot \left(\frac{zp \cdot 2\tau_{Pole\ pair}}{2\pi}\right)^2$	
Axis A	Axis B	Symbol	Description	Dimension unit
-	-	s	Length on which the specification for the number of encoder increments is based. (e.g. per pole distance or total length)	m
-	-	$2\tau_{pole\ pair}$	Pole distance of the permanent magnets, pole pair width	m
-	-	J	Moment of inertia = $J_{Forcer} + J_{Slide} + J_{Load}$	kg m^2
-	-	m	Moving mass = $m_{Forcer} + m_{Slide} + m_{Load}$	kg

0x2900 | 0x3100 - Speed controller: Parameter

Sub.	Name	Lenze setting	Data type
1	Speed controller: Gain	0.00033 Nm/rpm	UNSIGNED_32
2	Speed controller: Reset time	17.6 ms	UNSIGNED_16
3	Speed controller: Rate time	0.00 ms	UNSIGNED_16

Subindex 1: **Speed controller: Gain**				
Setting range (min. value | unit | max. value)			Lenze setting	
0.00000	Nm/rpm	20000.00000	0.00033 Nm/rpm	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/100000	UNSIGNED_32

Subindex 2: **Speed controller: Reset time**				
Setting range (min. value | unit | max. value)			Lenze setting	
1.0	ms	6000.0	17.6 ms	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	UNSIGNED_16

Subindex 3: **Speed controller: Rate time**				
Setting range (min. value | unit | max. value)			Lenze setting	
0.00	ms	3.00	0.00 ms	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/100	UNSIGNED_16

0x2901 | 0x3101 - Speed controller: Gain - adaptation

Setting range (min. value | unit | max. value)			Lenze setting	
0.00	%	200.00	100.00 %	
☑ Write access ☐ CINH ☑ OSC ☑ P ☑ RX ☐ TX			Scaling: 1/100	UNSIGNED_16

0x2902 | 0x3102 - Speed controller: Load starting value

Setting range (min. value | unit | max. value)			Lenze setting	
-1000.0	%	1000.0	0.0 %	
☑ Write access ☐ CINH ☑ OSC ☑ P ☑ RX ☐ TX			Scaling: 1/10	INTEGER_16

0x2903 | 0x3103 - Speed: Speed setpoint - filter time

Setting range (min. value | unit | max. value)			Lenze setting	
0.0	ms	50.0	0.0 ms	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	UNSIGNED_16

0x2904 | 0x3104 - Speed: Actual speed - filter time

Time constant for actual speed filter

- In order to maximise the dynamics of the speed control loop, the actual speed filter should be operated with a time constant as low as possible. The lower the time constant the higher the gain of the speed controller. Since actual value filters have the task to dampen measuring errors or interference components, it must be found a compromise between filter task and the resulting delay.
- When a Lenze motor is selected, a starting value is automatically preset for the time constant in order that most of the applications will be feasible. Modifications required for individual cases have to be determined empirically, e.g. due to
  - higher requirements in the drive dynamics,
  - lower quality of the encoder signals,
  - very long encoder cables.

Setting range (min. value | unit | max. value)			Lenze setting	
0.0	ms	50.0	0.6 ms	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	UNSIGNED_16

5.11.4 Setting the position controller

For the automatic calculation of the gain, the "Position controller: Calculate controller parameters" function is provided via object 0x2822 (or 0x3022 for axis B).

Equations for calculating the gain				
$V_p = \frac{1}{32 \cdot T_{Sum}}$			$T_{Sum} = T_{Filter} + T_{Current\ controller}$	
Axis A	Axis B	Symbol	Description	Dimension unit
0x2980	0x3180	V_p	Position controller: Gain	Hz
0x2904	0x3104	T_{filter}	Actual speed value filter time constant	s
-	-	$T_{current\ controller}$	Equivalent time constant of the current control loop = 0.0005 s (500 µs)	s

0x2980 | 0x3180 - Position controller: Gain

Setting range (min. value | unit | max. value)			Lenze setting	
0.00	1/s	1000.00	28.40 1/s	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/100	UNSIGNED_32

0x2981 | 0x3181 - Position controller: Gain - adaptation

Setting range (min. value | unit | max. value)			Lenze setting	
0.00	%	200.00	100.00 %	
☑ Write access ☐ CINH ☑ OSC ☑ P ☑ RX ☐ TX			Scaling: 1/100	UNSIGNED_16

0x2982 | 0x3182 - Position controller: Output signal limitation

Setting range (min. value | unit | max. value)			Lenze setting	
0	[n unit]	480000	480000 [n unit]	
☑ Write access ☐ CINH ☑ OSC ☑ P ☑ RX ☐ TX			Scaling: $480000/2^{31}$	UNSIGNED_32

0x2983 | 0x3183 - Position: Select a new actual position

Setting range (min. value | unit | max. value)			Lenze setting	
-2147483647	[Pos unit]	2147483647	0 [Pos unit]	
☑ Write access ☐ CINH ☑ OSC ☑ P ☑ RX ☐ TX				INTEGER_32

0x2984 | 0x3184 - Determine target position: Mode

Selection list (Lenze setting printed in bold)			
0	**Absolute: (target position = 0x2983)**		
1	Relative: (actual position = actual position + 0x2983)		
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_8

5.11.5 Setting the field controller (ASM)

For motors with great rotor time constants or small rotor resistances, very high gain factors are calculated. Since the setting range of the field controller is limited to the double rated magnetising current, the field control loop in the case of these motors tends to a two-point response when the values calculated are entered.

Tip!

Starting from a calculated gain factor of approx. 1000 A/Vs, do not set the full value anymore.

Example: Calculated value = 10000 A/Vs ➞ setting = 3000 A/Vs

For the automatic calculation of the controller parameters, the "Field controller: Calculate controller parameters" function is provided via object 0x2822 (or 0x3022 for axis B).

Equations for calculating the gain and the reset time

$$V_p \approx \frac{1}{4 \cdot R_r \cdot T_{\text{Current controller}}} \qquad \left(V_p = \frac{T_n}{4 \cdot K_{\text{Distance}} \cdot T_{\text{Current controller}}} = \frac{\frac{L_r}{R_r}}{4 \cdot L_h \cdot T_{\text{Current controller}}} \right)$$

$$T_n = T_r = \frac{L_r}{R_r}$$

Axis A	Axis B	Symbol	Description	Dimension unit
0x29C0:1	0x31C0:1	V_p	Field controller gain	A/Vs
0x29C0:2	0x31C0:2	T_n	Field contr. reset time	s
0x2C02:2	0x3402:2	L_h	Mutual motor inductance (ASM)	H
0x2C02:1	0x3402:1	R_r	Motor rotor resistance (ASM)	Ω
-	-	$T_{\text{current controller}}$	Equivalent time constant of the current control loop = 0.0005 s (500 µs)	s
-	-	T_r	Motor rotor time constant	
-	-	K_{path}	Gain of the controlled path	
-	-	L_r	Motor rotor inductance (ASM)	H

0x29C0 | 0x31C0 - Field controller: Parameter

Sub.	Name	Lenze setting	Data type
▸ 1	Field controller: Gain	165.84 A/Vs	UNSIGNED_32
▸ 2	Field controller: Reset time	15.1 ms	UNSIGNED_16

Subindex 1: **Field controller: Gain**				
Setting range (min. value | unit | max. value)			Lenze setting	
0.00	A/Vs	50000.00	165.84 A/Vs	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/100	UNSIGNED_32

Subindex 2: **Field controller: Reset time**				
Setting range (min. value | unit | max. value)			Lenze setting	
1.0	ms	6000.0	15.1 ms	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	UNSIGNED_16

5.11.6 Setting the field weakening controller (ASM)

Since the controlled system gain changes with the speed, the field weakening controller is corrected via the speed.

For the automatic calculation of the controller parameters, the "Field weakening controller: Calculate controller parameters" function is provided via object 0x2822 (or 0x3022 for axis B).

Equations for calculating the gain and the reset time

$$V_p = 0$$

$$V_{Distance\_Fs} = p \cdot n_{Transition} \cdot \frac{2\pi}{60}$$

$$T_n = 4 \cdot \frac{V_{Distance\_Fs}}{60} \cdot (T_{EF} + T_{Filter})$$

$$T_{EF} = T_r = \frac{L_r}{R_r} \approx \frac{L_h + L_{ss}}{R_r}$$

Axis A	Axis B	Symbol	Description	Dimension unit
0x29E0:1	0x31E0:1	V_p	Field weakening controller gain	Vs/V
-	-	V_{path_Fs}	Gain of the controlled path	
-	-	p	Number of pole pairs	
-	-	$n_{transition}$	Speed at which the field weakening is approximately initiated.	rpm
0x29E0:2	0x31E0:2	T_n	Field weak. contr. reset time	s
-	-	T_{EF}	Equivalent time constant of the field control loop	
0x29E3	0x31E3	T_{filter}	Filter time constant for the required voltage	s
-	-	T_r	Motor rotor time constant	
-	-	L_r	Motor rotor inductance	
0x2C02:2	0x3402:2	L_h	Mutual motor inductance (ASM)	H
0x2C01:3	0x3401:3	L_{ss}	Motor stator leakage inductance (ASM) or motor stator inductance (SM)	H
0x2C02:1	0x3402:1	R_r	Motor rotor resistance (ASM)	Ω

0x29E0 | 0x31E0 - Field weakening controller: Parameter

Sub.	Name	Lenze setting	Data type
1	Field weakening controller: Gain	0.000 Vs/V	UNSIGNED_32
2	Field weakening controller: Reset time	2000.0 ms	UNSIGNED_32

Subindex 1: **Field weakening controller: Gain**				
Setting range (min. value | unit | max. value)			Lenze setting	
0.000	Vs/V	2147483.647	0.000 Vs/V	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/1000	UNSIGNED_32

Subindex 2: **Field weakening controller: Reset time**				
Setting range (min. value | unit | max. value)			Lenze setting	
1.0	ms	240000.0	2000.0 ms	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	UNSIGNED_32

0x29E1 | 0x31E1 - field: Field set value limitation

Setting range (min. value | unit | max. value)			Lenze setting	
5.00	%	100.00	100.00 %	
☑ Write access ☐ CINH ☑ OSC ☑ P ☑ RX ☐ TX			Scaling: 1/100	UNSIGNED_16

0x29E2 | 0x31E2 - DC link circuit voltage: Filter time

Setting range (min. value | unit | max. value)			Lenze setting	
1.0	ms	1000.0	25.0 ms	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	UNSIGNED_16

0x29E3 | 0x31E3 - motor: Actual voltage - filter time

Setting range (min. value | unit | max. value)			Lenze setting	
1.0	ms	1000.0	25.0 ms	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	UNSIGNED_16

0x29E4 | 0x31E4 - Voltage reserve range

From version 01.03

Voltage reserve range at the transition point to the field weakening

- Only relevant for servo control for asynchronous motor (selection "2" in 0x2C00 or 0x3400 for axis B).

Setting range (min. value | unit | max. value)			Lenze setting	
1	%	20	5 %	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_8

5.11.7 Field weakening operation - synchronous motor (SM)

The control of the i700 servo inverter permits the operation of a synchronous motor outside the voltage range. If a motor is selected in the »Easy Starter«, the control is automatically parameterised.

In order to improve the transition from the base speed range to field weakening, we recommend the activation of the current controller feedforward control via the object 0x2941 (or 0x3141 for axis B).

- The current controller feedforward control is determined via the following parameters:
  - Motor stator resistance (0x2C01:2 or 0x3401:2 for axis B)
  - Motor stator leakage inductance (0x2C01:3 or 0x3401:3 for axis B)
  - e.m.f constant (0x2C03:1 or 0x3403:1 for axis B)
- If a third-party motor is to be operated in the field weakening range, the parameters mentioned before have to be selected carefully.

Stop!

Operation of synchronous motors outside the voltage range:

If a pulse inhibit is set in the controller, e.g. in the case of a controller inhibit or in the event of an error, the DC bus will load up to the terminal voltage corresponding to the actual speed (see equation below).

- At high speeds outside the voltage range, the terminal voltage may be higher than the mains voltage!
- Connect a brake chopper to the DC bus to prevent the DC bus from reaching impermissibly high voltages!

The terminal voltage approximately corresponds to the following equation:

Equations for calculating the terminal voltage

$$V_{Terminal} = \text{Actual speed} \cdot \frac{\text{Rated mains voltage}}{\text{Rated motor speed}}$$

Axis A	Axis B	Description	Dimension unit
0x2540:1		Rated mains voltage	V
0x2C01:4	0x3401:4	Rated motor speed	r/min

5.12 Fine adjustment des motor model

The further commissioning steps are only required for servo controls if more stringent requirements with regard to the torque linearity have to be met. During the commissioning process of Lenze motors, typical values for the relevant parameters are provided. For motors of other manufacturers, these values are to be requested from the motor manufacturer, or they have to be estimated.

5.12.1 Correction of the stator leakage inductance (Lss)...

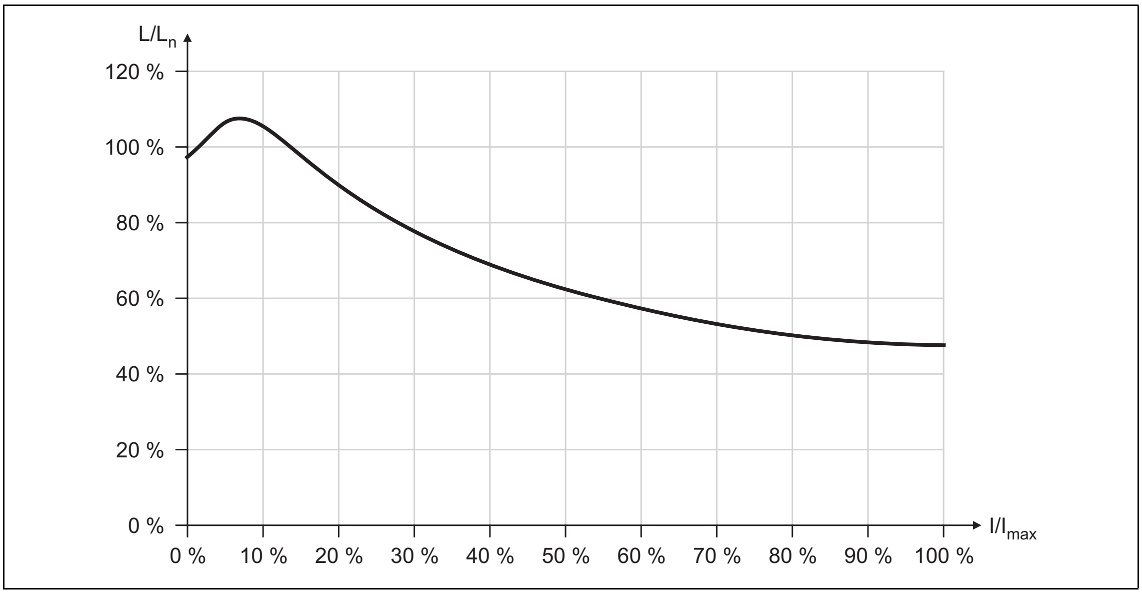
...and the current controller parameters by means of the saturation characteristic

For the most part, the electrical characteristics of the motor are the relevant factors for an optimal current controller setting (Vp, Ti), especially the stator resistance and the stator leakage inductance (Lss). However, modern motors have their stator leakage inductance changed along with the current level so that it is impossible to have an optimal current controller setting for all working points at all times.

For applications with operating phases that involve very different current and torque requirements and, at the same time, high requirements on dynamic drive behaviour, the i700 servo inverter provides the possibility of the correction of the stator leakage inductance and the current controller settings by means of the adjustable saturation characteristic.

The saturation characteristic is a typical characteristic of motors of one type/size. It does not depend on the maximum process current of the motor in the prevailing application. Thus the defined values should be based on the key data of the motors. These are rated motor current, peak motor current for a limited time and the ultimate motor current.

The following picture shows a typical saturation characteristic of an MCS motor:



[5-1] Saturation characteristic: Inductance referring to the inductance for rated current

The saturation characteristic represents the change in inductance (L/Ln) as a function of the motor current (I/Imax). The variables of both axes which were scaled to a reference value are represented as percentages.

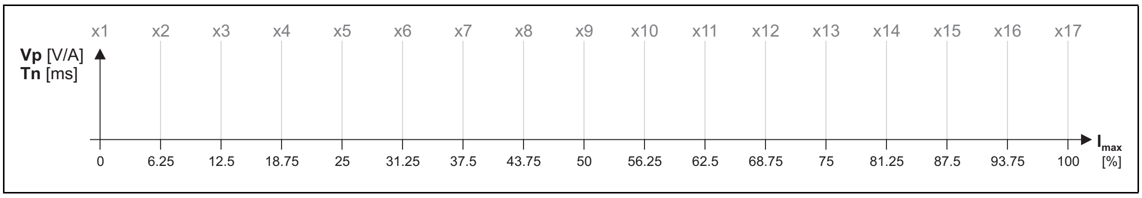
- When a Lenze motor is selected, the saturation characteristic is already filled with values typical of the series.

Note!

The saturation characteristic is not only used to correct the current controller, but it also influences the current controller feedforward control (can be activated via object 0x2941 or 0x3141 for axis B).

Distribution of the grid points

- The saturation characteristic is represented by means of 17 grid points.
- The 17 grid points are spaced on the X axis at equal intervals (equidistantly) in a range of 0 ... 100 %. The 100% value of the X axis refers to the current value (max. motor current in the process) set in object 0x2C05 (or 0x3405 for axis B).
- The Y values for the grid points can be accessed via the subindices of object 0x2C04 (or 0x3404 for axis B).



[5-2] Saturation characteristic: Distribution of the grid points

- The 100 % value of a grid point refers to
  - the set motor stator leakage inductance (0x2C01:3 or 0x3401:3 for axis B) and
  - the set current controller gain Vp (0x2942:1 or 0x3142:1 for axis B).
- Preferably select a display area of the grid points which includes at least the ultimate motor current. The current controller step response is then recorded actively only until the grid point with peak motor current. In order to prevent the motor winding from being overloaded, the manual test mode "current pulse" should be used for recording. The grid points with current setpoints above the peak motor current are determined through interpolation.
- When the saturation characteristics for motor types are determined, it makes sense in some cases to select a scaled representation of the grid point distribution. This requires to know the highest value of the quotient from "ultimate motor current / rated motor current" of the motor series.

0x2C04 | 0x3404 - Motor: Lss saturation characteristic - inductance grid points (y)

From version 01.05, the correction by means of saturation characteristic via the subindex 18 can also be switched off.

Sub.	Name	Lenze setting	Data type
1	Lss: y1 = L01 (x = 0.00 %)	165 %	UNSIGNED_16
2	Lss: y2 = L02 (x = 6.25 %)	200 %	UNSIGNED_16
3	Lss: y3 = L03 (x = 12.50 %)	146 %	UNSIGNED_16
4	Lss: y4 = L04 (x = 18.75 %)	117 %	UNSIGNED_16
5	Lss: y5 = L05 (x = 25.00 %)	97 %	UNSIGNED_16
6	Lss: y6 = L06 (x = 31.25 %)	82 %	UNSIGNED_16
7	Lss: y7 = L07 (x = 37.50 %)	71 %	UNSIGNED_16
8	Lss: y8 = L08 (x = 42.75 %)	62 %	UNSIGNED_16
9	Lss: y9 = L09 (x = 50.00 %)	55 %	UNSIGNED_16
10	Lss: y10 = L10 (x = 56.25 %)	50 %	UNSIGNED_16
11	Lss: y11 = L11 (x = 62.50 %)	46 %	UNSIGNED_16
12	Lss: y12 = L12 (x = 68.75 %)	43 %	UNSIGNED_16
13	Lss: y13 = L13 (x = 75.00 %)	42 %	UNSIGNED_16
14	Lss: y14 = L14 (x = 81.25 %)	41 %	UNSIGNED_16
15	Lss: y15 = L15 (x = 87.50 %)	41 %	UNSIGNED_16
16	Lss: y16 = L16 (x = 93.25 %)	41 %	UNSIGNED_16
17	Lss: y17 = L17 (x = 100.00 %)	41 %	UNSIGNED_16
18	Motor: Lss saturation characteristic - Activation	1: Active	UNSIGNED_16
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			

0x2C05 | 0x3405 - Motor: Lss saturation charcteristic - reference for current grid points (x)

Maximum motor current
- Serves as a reference value for the scaled current values of the X axis of the saturation characteristic.

Setting range (min. value | unit | max. value)			Lenze setting	
0.0	A	500.0	5.4 A	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	UNSIGNED_16

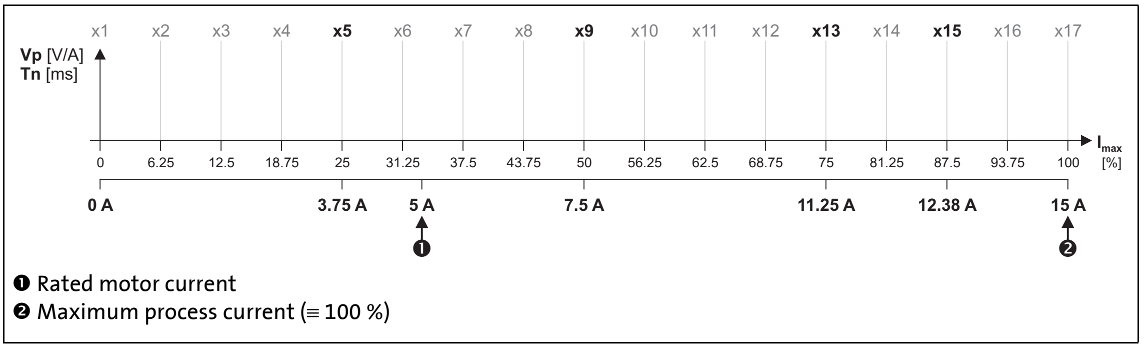
5.12.1.1 Example for determining the saturation characteristic

Given values:

- Rated motor current: 5 A
- Maximum motor current: 20 A
- Maximum process current: 15 A

Procedure:

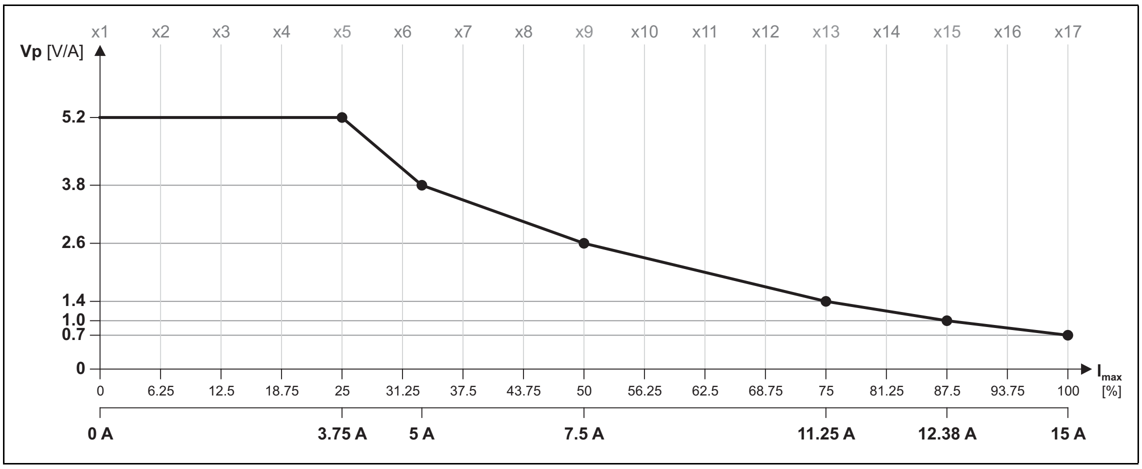
1. Deactivate correction: Set all subindices of object 0x2C04 (or 0x3404 for axis B) to 100 %.
2. Use object 0x2C05 (or 0x3405 for axis B) to set the maximum current up to which the motor is to be actuated in the process (in this example "15 A").
3. Adjust the current controller with different current setpoints by means of the manual test mode "current pulse" and take down the corresponding settings for Vp and Tn.
   - The procedure is described in the Manual test mode "Current pulse" chapter. (📖 108)
   - The current setpoints to be set for the corresponding adjustment in object 0x2835:1 (or 0x3035:1 for axis B) result from the scaling of the maximum process current to the X axis of the saturation characteristic.
   - The grid points which are required to define the saturation characteristic with a sufficient quality varies from motor to motor and thus has to be determined individually.
   - For this example, currents that are part of the grid points 5, 9, 13, and 15 have been selected, and a measurement at rated motor current was carried out additionally:



[5-3] Saturation characteristic: Distribution of the grid points

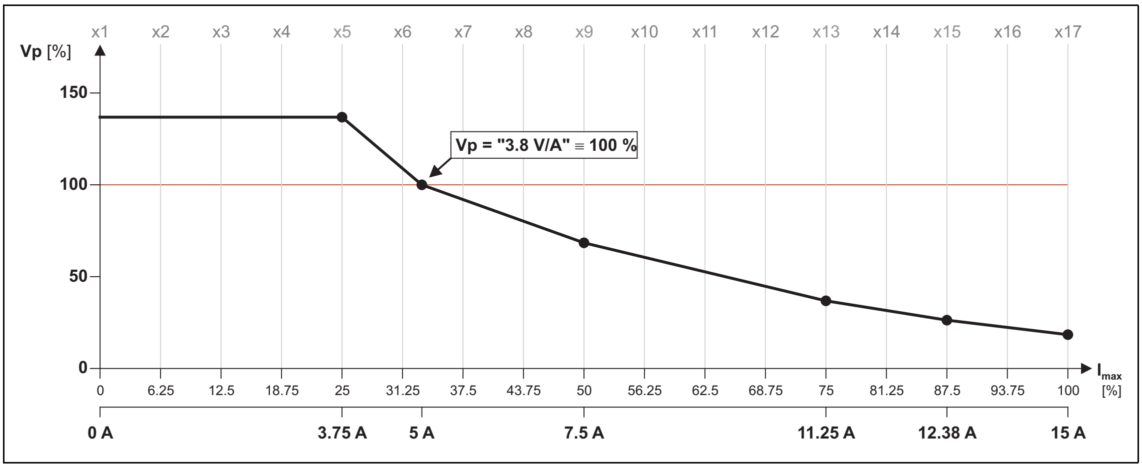
Specifications for adjustment			Measured values	
Grid point	Scaling	Current setpoint	Vp [V/A]	Tn [ms]
5	0.25 * 15 A =	3.75 A	5.2	6.5
9	0.5 * 15 A =	7.5 A	2.6	4
13	0.75 * 15 A =	11.25 A	1.4	2.5
15	0.875 * 15 A =	12.38 A	1.0	2
17	1.0 * 15 A =	15 A	0.7	1.7
	Rated motor current =	5 A	3.8	5

4. Set the characteristic by means of the determined values for Vp (but do not enter any values in 0x2C04 or 0x3404 for axis B yet).
   - Here, the values of the grid points which have not been adjusted must be determined by interpolation between two values.
   - **Note:** In this example it was assumed that the inductance does not change considerably below 3.75 A. For this reason the same Vp value resulting from a measurement with a motor current of 3.75 A was used for all grid points below 3.75 A.



[5-4] Determined saturation characteristic

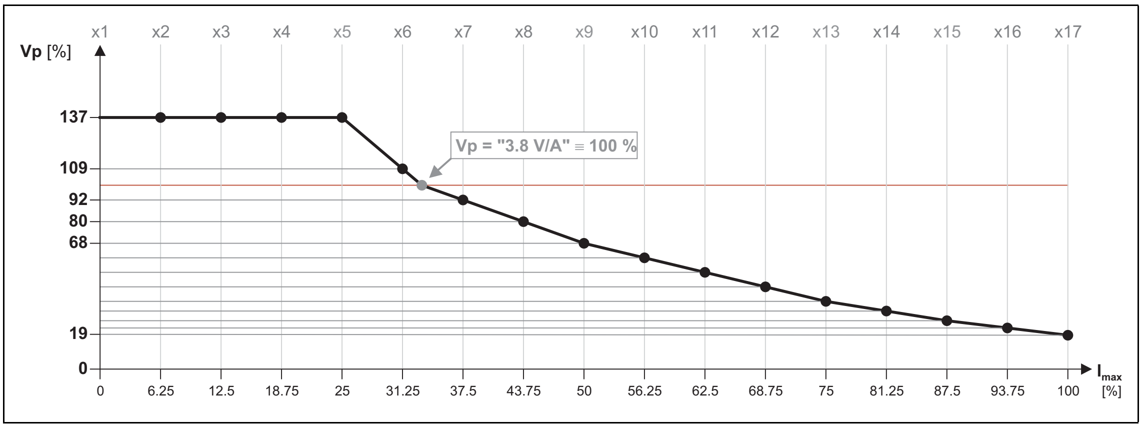
5. Set the gain Vp and the reset time Tn to the values that have been determined during the adjustment with rated motor current (in this example "5 A"):
   - Set 0x2942:1 (or 0x3142:1 for axis B) = "3.8 V/A".
   - Set 0x2942:2 (or 0x3142:2 for axis B) = "5 ms".
6. Scale the Vp values on the Y axis of the characteristic to a Vp setting of "3.8 V/A":



[5-5] Scaling of the determined saturation characteristic to "100 % Vp"

7. Enter the Vp values in per cent from the grid points into the subindices of object 0x2C04 (or 0x3404 for axis B):

Setting for grid points 1 ... 17 in [%]																
y1	y2	y3	y4	y5	y6	y7	y8	y9	y10	y11	y12	y13	y14	y15	y16	y17
137	137	137	137	137	109	92	80	68	61	53	45	37	32	26	22	19



[5-6] Grid point values of the saturation characteristic determined

8. Enter the maximum process current ("15 A") as maximum current in object 0x6073 (or 0x6873 for axis B).
   - With these settings, the same current characteristic should occur, irrespective of the current magnitude.
   - Since the current controller gain is now corrected actively, the step responses may differ slightly compared to the previous measurements. In this case, the current controller parameters must be optimised one last time.
9. For permanent storage: Upload the detected characteristic from the i700 servo inverter into the Controller.

   The »EASY Starter« serves to save the parameter settings of the i700 servo inverter as parameter file (*.gdc). In the »PLC Designer«, this file can then be imported in the corresponding axis.

   ▶ Saving changed parameters safe against mains failure (📖 60)

5.12.2 Synchronous motor (SM): Compensating for temperature and current influences

The properties of the permanent magnets of permanently excited synchronous motors (SM, PSM) depend on the temperature and the amperage. The relationship between motor current and resulting torque changes correspondingly.

The influences of the temperature and the amperage on the magnetisation can be taken into account by the motor control and hence be compensated for.

- To compensate for the temperature dependence of the magnets, the temperature coefficient (kT) of the permanent magnet must be entered in object 0x2C03:3 (or 0x3403:3 for axis B) (linear characteristic).
- To compensate for the current dependence of the magnets, multiple grid points of a characteristic must be entered in the following object (non-linear characteristic):

0x2C06 | 0x3406 - Motor (SM): Magnet characteristic (current) - grid points

Sub.	Name	Lenze setting	Data type
1	Magnet characteristic: x1 = i01/iN	0 %	UNSIGNED_16
2	Magnet characteristic: y1 = kT01/kTN	100 %	UNSIGNED_16
3	Magnet characteristic: x2 = i02/iN	100 %	UNSIGNED_16
4	Magnet characteristic: y2 = kT02/kTN	100 %	UNSIGNED_16
5	Magnet characteristic: x3 = i03/iN	200 %	UNSIGNED_16
6	Magnet characteristic: y3 = kT03/kTN	100 %	UNSIGNED_16
7	Magnet characteristic: x4 = i04/iN	415 %	UNSIGNED_16
8	Magnet characteristic: y4 = kT04/kTN	72 %	UNSIGNED_16
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			

5.12.3 Asynchronous motor (ASM): Identifying the Lh saturation characteristic

In case of an asynchronous motor, the relationship between current and torque is basically determined by the saturation behaviour of the mutual inductance. If the achieved torque accuracy, especially in the field weakening range should not be sufficient, the accuracy can be increased by the individual identification of the saturation characteristic. This behaviour can be measured by the i700 servo inverter.

Preconditions for the execution

- Before this commissioning function is executed, the inverter characteristic and the motor parameters have to be identified.
  ▶ Compensating for inverter influence on output voltage (📖 71)
  ▶ Determine motor parameters automatically via "motor parameter identification" (📖 78)
- The motor may be firmly braked.
- The controller is free of errors and is in the "Switched on" device status.

Response of the motor during the execution

Standstill

How to identify the LH saturation characteristic:

1. If the controller is enabled, inhibit the controller.
   ▶ Enable/inhibit operation via control word (📖 59)
2. Set object 0x2825 (or 0x3025 for axis B) to "10" to change to the "Determine Lh saturation characteristicn" operating mode.
3. Enable the controller to start the procedure.

   Notes:
   - The identification of the Lh saturation characteristic can take up to 11 minutes. The progress can be checked in the object 0x2823 (or 0x3023 for axis B).
   - By means of controller inhibit, the started procedure can be cancelled any time, if required. Characteristic values that have already been determined are rejected in this case.

After successful completion...

...the controller will be inhibited automatically and the points of the determined LH characteristic will be set in object 0x2C07 (or 0x3407 for axis B).

- For permanent storage, the changed settings must be uploaded to the controller from the i700 servo inverter.
  The »EASY Starter« serves to save the parameter settings of the i700 servo inverter as parameter file (*.gdc). In the »PLC Designer«, this file can then be imported in the corresponding axis.
  ▶ Saving changed parameters safe against mains failure (📖 60)
- The controller inhibit automatically set by the procedure can be deactivated via the Controlword (0x6040 or 0x6840 for axis B) (setting = 7, 15).

In the event of an error

If an error occurs during the procedure or the pulse inhibit gets active (e.g. due to short-time undervoltage), the procedure is terminated with controller inhibit without a change in settings.

Loading the standard Lh saturation characteristic

If an incorrect Lh saturation characteristic has been determined or none at all, it is possible to load a device-typical standard Lh characteristic.

How to load the standard Lh saturation characteristic:

1. Set object 0x2822 (or 0x3022 for axis B) to "13".
   - The progress of the procedure is shown in object 0x2823 (or 0x3023 for axis B).
2. For permanent storage: After completion of the procedure, upload the Lh saturation characteristic set in 0x2C07 (or 0x3407 for axis B) to the controller from the i700 servo inverter.

   The »EASY Starter« serves to save the parameter settings of the i700 servo inverter as parameter file (*.gdc). In the »PLC Designer«, this file can then be imported in the corresponding axis. ▶ Saving changed parameters safe against mains failure (📖 60)

0x2C07 | 0x3407 - Motor (ASM): Lh saturation characteristic - inductance grid points (y)

Sub.	Name	Lenze setting	Data type
1	Lh: y1 = L01 (x = 0.00 %)	118 %	UNSIGNED_16
2	Lh: y2 = L02 (x = 6.25 %)	118 %	UNSIGNED_16
3	Lh: y3 = L03 (x = 12.50 %)	118 %	UNSIGNED_16
4	Lh: y4 = L04 (x = 18.75 %)	117 %	UNSIGNED_16
5	Lh: y5 = L05 (x = 25.00 %)	116 %	UNSIGNED_16
6	Lh: y6 = L06 (x = 31.25 %)	114 %	UNSIGNED_16
7	Lh: y7 = L07 (x = 37.50 %)	111 %	UNSIGNED_16
8	Lh: y8 = L08 (x = 43.75 %)	107 %	UNSIGNED_16
9	Lh: y9 = L09 (x = 50.00 %)	100 %	UNSIGNED_16
10	Lh: y10 = L10 (x = 56.25 %)	93 %	UNSIGNED_16
11	Lh: y11 = L11 (x = 62.50 %)	86 %	UNSIGNED_16
12	Lh: y12 = L12 (x = 68.75 %)	78 %	UNSIGNED_16
13	Lh: y13 = L13 (x = 75.00 %)	71 %	UNSIGNED_16
14	Lh: y14 = L14 (x = 81.25 %)	64 %	UNSIGNED_16
15	Lh: y15 = L15 (x = 87.50 %)	57 %	UNSIGNED_16
16	Lh: y16 = L16 (x = 93.75 %)	50 %	UNSIGNED_16
17	Lh: y17 = L17 (x = 100.00 %)	42 %	UNSIGNED_16
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			

5.12.4 Estimating the optimal magnetising current

In case of the given Lh saturation behaviour, there is (usually) a magnetising current where the torque efficiency is highest. This magnetising current can be determined by the i700 servo inverter.

- The execution of this function also changes the Lh saturation characteristic (0x2C07 or 0x3407 for axis B) (compressed, extended).
- After executing the function, the magnetising current determined is entered in object 0x2C02:3 (or 0x3402:3 for axis B).

Preconditions for the execution

- Before this commissioning function is executed, the motor parameters and the Lh saturation characteristic have to be identified.
  - Determine motor parameters automatically via "motor parameter identification" (📖 78)
  - Asynchronous motor (ASM): Identifying the Lh saturation characteristic (📖 129)
- The motor may be firmly braked.

Response of the motor during the execution

Standstill

How to estimate the optimal magnetising current:

1. Set object 0x2822 (or 0x3022 for axis B) to "2".
   - The progress of the procedure is shown in object 0x2823 (or 0x3023 for axis B).
2. For permanent storage: Upload the changed controller parameters to the controller from the i700 servo inverter after the procedure has been completed:
   - Lh saturation characteristic (0x2C07 or 0x3407 for axis B)
   - Magnetising current (0x2C02:3 or 0x3402:3 for axis B)

   The »EASY Starter« serves to save the parameter settings of the i700 servo inverter as parameter file (*.gdc). In the »PLC Designer«, this file can then be imported in the corresponding axis. Saving changed parameters safe against mains failure (📖 60)

5.13 Parameterising filter elements in the setpoint path

5.13.1 Jerk limitation

Max. acceleration change

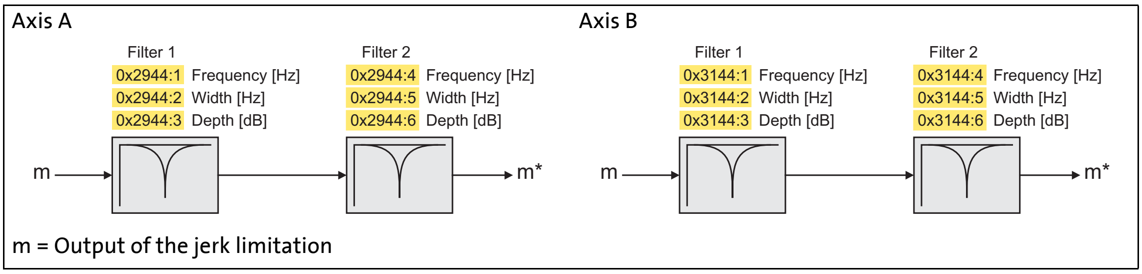
0x2945 | 0x3145 - Torque: Setpoint jerk limitation

Setting range (min. value | unit | max. value)			Lenze setting	
0.1	%	400.0	400.0 %	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	UNSIGNED_16

5.13.2 Notch filters (band-stop filters)

Due to the high dynamic performance or the high limit frequency of the closed current control loop, mechanical natural frequencies can be excited, which can result in resonance and thus cause the speed control loop to become unstable.

In order to suppress or damp these resonant frequencies, two notch filters are integrated in the speed control loop of the controller, which can be parameterised. In the Lenze setting, these filters are switched off:



[5-7] Optional notch filters (filter cascade) in the speed control loop

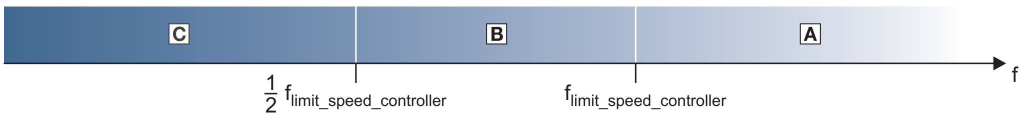
Use of the notch filters depending on the resonant frequency

Stop!

If the filter parameters are set incorrectly, the impaired closed-loop control can respond with too large overshoots and cause the controller to become unstable, e.g. if the filter width is set to a value more than twice as large as the filter frequency.

After setting the filter parameters, the drive behaviour for quick stop (QSP) must be checked.

In the case of impairment,
- the drive that is still running must either be coasted down by activating the controller inhibit or immediately be brought to a standstill via a brake.
- the speed controller must be optimised again afterwards.
- the test procedure must be repeated.



A Usual and intended use of the notch filter
B Use of the notch filter is only partially suitable
C The notch filter should not be used

[5-8] Use of the notch filter depending on the resonant frequency

- The notch filters are suitable for use with resonant frequencies equal to or higher than the limit frequency of the speed controller.
  - Resonant frequencies ≥ $f_{limit\_speed\_controller}$ = 70 Hz ... 110 Hz
- For resonant frequencies that are lower than the limit frequency of the speed controller, the use of suitable speed profiles with an S-shaped ramp is recommended.

Setting the notch filters

Since the exact frequency response of the speed control path in most cases is not known beforehand, an experimental procedure for setting the notch filters is described in the following.

How to set the notch filters:

1. Setting and optimising the current controller.
2. Adapt the speed controller reset time to the filter time constant of the speed filter and the equivalent time constant of the current control loop:
   - The following applies to axis A: 0x2900:2 = 16 * (0x2904 + 500 µs)
   - The following applies to axis B: 0x3100:2 = 16 * (0x3104 + 500 µs)

   Note: The setting of the reset time includes the equivalent time constant of the current control loop. The 500 µs indicated are typical in a power range of up to 50 kW. Above this value, greater time constants may occur.
3. Slowly increase the proportional gain of the speed controller in 0x2900:1 (or 0x3100:1 for axis B) until the speed control loop starts to be unstable (acoustic determination or measurement of the motor current).
4. Measure the oscillation frequency using an oscilloscope (observe current or speed).
5. Set the oscillation frequency determined as filter frequency in 0x2944:1 (or 0x3144:1 for axis B).
6. Set the filter width to 50 % of the filter frequency in 0x2944:2 (or 0x3144:2 for axis B).
   - Example: filter frequency = 200 Hz → filter width = 100 Hz.
7. Set the filter depth to 40 dB in 0x2944:3 (or 0x3144:3 for axis B).
   - If "0 dB" are set (default setting), the filter is not effective.
8. Further increase proportional gain of the speed controller in 0x2900:1 (or 0x3100:1 for B) until the speed control loop starts to be unstable again.
   - If the oscillation frequency has changed now, readjust the filter frequency by trimming. The use of a second filter is ineffective here.
   - If the oscillation frequency remains the same, readjust the filter depth and/or the filter width by trimming (the first reduces the amplitude, the second lets the phase rotate faster).
   - Repeat step 8 until the desired behaviour or the limit of a sensible speed controller gain has been reached.

Note!

Readjust the speed controller after setting the notch filters. ▶ Setting the speed controller. (📖 113)

For permanent storage, the changed settings must be uploaded to the controller from the i700 servo inverter.

The »EASY Starter« serves to save the parameter settings of the i700 servo inverter as parameter file (*.gdc). In the »PLC Designer«, this file can then be imported in the corresponding axis. ▶ Saving changed parameters safe against mains failure (📖 60)

0x2944 | 0x3144 - Torque: Notch filter setpoint torque

Sub.	Name	Lenze setting	Data type
▶ 1	Notch filter 1: Frequency	200.0 Hz	UNSIGNED_16
▶ 2	Notch filter 1: Bandwidth	20.0 Hz	UNSIGNED_16
▶ 3	Notch filter 1: Damping	0 db	UNSIGNED_8
▶ 4	Notch filter 2: Frequency	400.0 Hz	UNSIGNED_16
▶ 5	Notch filter 2: Bandwidth	40.0 Hz	UNSIGNED_16
▶ 6	Notch filter 2: Damping	0 db	UNSIGNED_8

Subindex 1: **Notch filter 1: Frequency**				
Setting range (min. value | unit | max. value)			Lenze setting	
1.0	Hz	1000.0	200.0 Hz	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	UNSIGNED_16

Subindex 2: **Notch filter 1: Bandwidth**				
Setting range (min. value | unit | max. value)			Lenze setting	
0.0	Hz	500.0	20.0 Hz	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	UNSIGNED_16

Subindex 3: **Notch filter 1: Damping**				
Setting range (min. value | unit | max. value)			Lenze setting	
0	dB	100	0 db	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_8

Subindex 4: **Notch filter 2: Frequency**				
Setting range (min. value | unit | max. value)			Lenze setting	
1.0	Hz	1000.0	400.0 Hz	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	UNSIGNED_16

Subindex 5: **Notch filter 2: Bandwidth**				
Setting range (min. value | unit | max. value)			Lenze setting	
0.0	Hz	500.0	40.0 Hz	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	UNSIGNED_16

Subindex 6: **Notch filter 2: Damping**				
Setting range (min. value | unit | max. value)			Lenze setting	
0	dB	100	0 db	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_8

5.14 Parameterising the V/f characteristic control

In case of the V/f characteristic control (VFCplus), the motor voltage of the inverter is determined by means of a linear or quadratic characteristic depending on the field frequency or motor speed to be generated. The voltage follows a preselected characteristic.

Stop!

- The V/f characteristic control is only suitable for asynchronous motors.
- Observe the following when you actuate drives with a square-law V/f characteristic:
  - Please always check whether the corresponding drive is suitable for operation with a quadratic V/f characteristic!
  - If your pump drive or fan drive is not suitable for operation with a square-law V/f characteristic, use the linear V/f characteristic instead, or select the servo control.
- For adjustment, observe the thermal performance of the connected asynchronous motor at low output frequencies.
  - Usually, standard asynchronous motors with insulation class B can be operated for a short time with their rated current in the frequency range 0 Hz ... 25 Hz.
  - Contact the motor manufacturer to get the exact setting values for the max. permissible motor current of self-ventilated motors in the lower speed range.
  - For square-law V/f characteristics we recommend setting a smaller V_{min}.
- As regards the motor nameplate data, at least the rated speed (0x2C01:4 or 0x3401:4 for axis B) and the rated frequency (0x2C01:5 or 0x3401:5 for axis B) must be entered for the i700 servo inverter to calculate the correct number of pole pairs.

Initial commissioning steps

After the motor and controller have been optimally adjusted to each other, the following "initial commissioning steps" are sufficient for a simple V/f characteristic control.

1. Defining the V/f characteristic shape
2. Activating the voltage vector control (Imin controller)
   – or alternatively –
   Setting the voltage boost
3. Setting the load adjustment
4. Defining the behaviour at the current limit (Imax controller)

Optimising the control mode

The following "optimisation steps" can be used to further optimise the control mode of the V/f characteristic control and adapt it to the practical application:

1. Setting the slip compensation
2. Setting the oscillation damping
3. Manual test mode "Current pulse"
   - Only required if the voltage vector control is applied, or if DC-injection braking or the flying restart process is activated.

Parameterisable functions

Optionally the following functions can be activated for the V/f characteristic control:

- "DC-injection braking" function
- "Flying restart" function

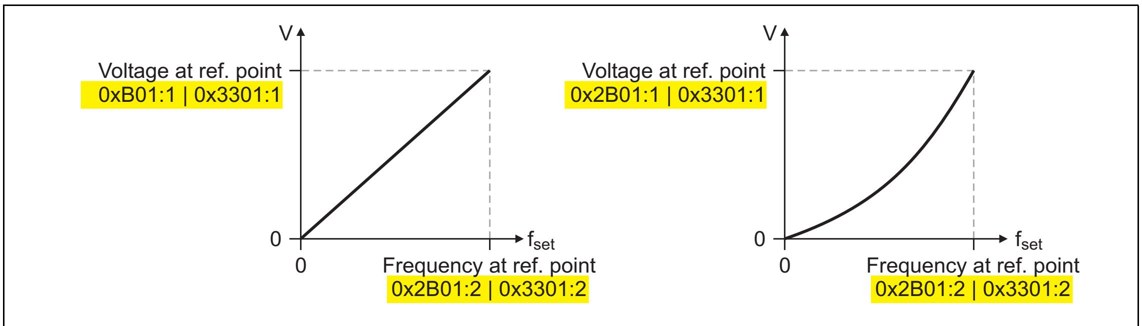
In the Lenze setting these two functions are deactivated.

5.14.1 Defining the V/f characteristic shape

For purposes of adaption to different load profiles, the shape of the characteristic can be selected:

0x2B00 | 0x3300 - VFC: V/f characteristic - shape

Selection list (Lenze setting printed in bold)		Info
0	**Linear (standard)**	Linear characteristic for drives with constant load torque over the speed.
1	Square-law (pumps and fans)	Square-law characteristic for drives with a linear or square-law load torque over the speed. • Square-law V/f characteristics are preferably used for centrifugal pumps and fan drives. • Please always check whether the corresponding drive is suitable for operation with a quadratic V/f characteristic! • If your pump drive or fan drive is not suitable for operation with a square-law V/f characteristic, use the linear V/f characteristic instead, or select the servo control.
2	User-defined	User-definable V/f characteristic for drives requiring an adaptation of the magnetising current via the output speed. • The user-definable V/f characteristic can for instance be used for operation on special machines like the reluctance motor, for the suppression of oscillations on the machine, or for energy optimisation.
☑ Write access ☑ CINH ☐ OSC ☑ P ☐ RX ☐ TX		UNSIGNED_8



[5-9] Principle of a linear V/f characteristic (on the left) and a quadratic V/f characteristic (on the right)

0x2B01 | 0x3301 - VFC: V/f characteristic - define reference point

Sub.	Name	Lenze setting	Data type
1	VFC: V/f characteristic - voltage at reference point	225 V	UNSIGNED_16
2	VFC: V/f characteristic - frequency at reference point	270 Hz	UNSIGNED_16

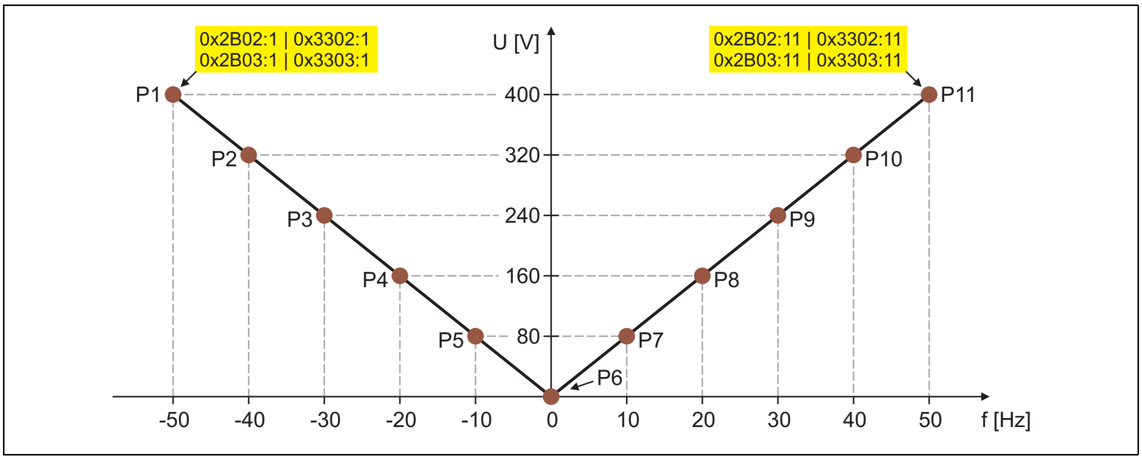
Subindex 1: **VFC: V/f characteristic - voltage at reference point**				
Setting range (min. value | unit | max. value)			Lenze setting	
0	V	5000	225 V	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

Subindex 2: **VFC: V/f characteristic - frequency at reference point**				
Setting range (min. value | unit | max. value)			Lenze setting	
0	Hz	5000	270 Hz	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

Defining a user-defined V/f characteristic

The "User-definable V/f characteristic" is provided for the individual adjustment of the motor magnetisation to the actual application if linear and square-law characteristics are not suitable.

- The characteristic is defined by means of 11 parameterisable grid points (voltage/frequency values).
- In the Lenze setting the 11 grid points represent a linear characteristic:



	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
V	400 V	320 V	240 V	160 V	80 V	0 V	80 V	160 V	240 V	320 V	400 V
f	-50 Hz	-40 Hz	-30 Hz	-20 Hz	-10 Hz	0 Hz	10 Hz	20 Hz	30 Hz	40 Hz	50 Hz

[5-10] User-definable V/f characteristic (Lenze setting)

0x2B02 | 0x3302 - VFC: User-definable V/f characteristic - frequency grid points (x)

Sub.	Name	Lenze setting	Data type
1	V/f: x1 = f01	-50 Hz	INTEGER_16
2	V/f: x2 = f02	-40 Hz	INTEGER_16
3	V/f: x3 = f03	-30 Hz	INTEGER_16
4	V/f: x4 = f04	-20 Hz	INTEGER_16
5	V/f: x5 = f05	-10 Hz	INTEGER_16
6	V/f: x6 = f06	0 Hz	INTEGER_16
7	V/f: x7 = f07	10 Hz	INTEGER_16
8	V/f: x8 = f08	20 Hz	INTEGER_16
9	V/f: x9 = f09	30 Hz	INTEGER_16
10	V/f: x10 = f10	40 Hz	INTEGER_16
11	V/f: x11 = f11	50 Hz	INTEGER_16
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			

0x2B03 | 0x3303 - VFC: User-definable V/f characteristic - voltage grid points (y)

Sub.	Name	Lenze setting	Data type
1	V/f: y1 = U01 (x = f01)	400.00 V	UNSIGNED_32
2	V/f: y2 = U02 (x = f02)	320.00 V	UNSIGNED_32
3	V/f: y3 = U03 (x = f03)	240.00 V	UNSIGNED_32
4	V/f: y4 = U04 (x = f04)	160.00 V	UNSIGNED_32
5	V/f: y5 = U05 (x = f05)	80.00 V	UNSIGNED_32
6	V/f: y6 = U06 (x = f06)	0.00 V	UNSIGNED_32
7	V/f: y7 = U07 (x = f07)	80.00 V	UNSIGNED_32
8	V/f: y8 = U08 (x = f08)	160.00 V	UNSIGNED_32
9	V/f: y9 = U09 (x = f09)	240.00 V	UNSIGNED_32
10	V/f: y10 = U10 (x = f10)	320.00 V	UNSIGNED_32
11	V/f: y11 = U11 (x = f11)	400.00 V	UNSIGNED_32
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			

5.14.2 Activating the voltage vector control (Imin controller)

The voltage vector control that can be activated is used if a comparatively high starting torque has to be ensured. The voltage vector control makes sure that the motor current required is kept at smaller speeds in the speed range.

Note!

The voltage vector control acts additively to the voltage boost.
- Only use one of the two "Boost" functions.
  Recommendation: voltage vector control

If the voltage vector control is used, greater losses and therefore an increased heating of the motor result from the increased current at low speeds.

- The voltage vector control is activated by the selection of a current setpoint.
- For the automatic calculation of the control parameters, the "Calculate Imin controller" function is provided via object 0x2822 (or 0x3022 for axis B).

0x2B04 | 0x3304 - VFC: Voltage vector control - setpoint current

Current setpoint for voltage vector control
- If "0.00 A" are set, the voltage vector control is deactivated.
- When the current setpoint is defined, provide a reserve of 20 % to prevent a motor stalling caused by sudden additional loads.
- Example for starting torque = rated motor torque:
  The current setpoint must be set to approx. 120 % of the load current.

Setting range (min. value | unit | max. value)			Lenze setting	
0.00	A	500.00	0.00 A	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/100	UNSIGNED_32

0x2B05 | 0x3305 - VFC: Voltage vector control parameter

Sub.	Name	Lenze setting	Data type
1	VFC: Voltage vector controller - gain	148.21 V/A	UNSIGNED_32
2	VFC: Voltage vector controller - reset time	3.77 ms	UNSIGNED_32

Subindex 1: **VFC: Voltage vector controller - gain**				
Gain for voltage vector control				
Setting range (min. value | unit | max. value)			Lenze setting	
0.00	V/A	750.00	148.21 V/A	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/100	UNSIGNED_32

Subindex 2: **VFC: Voltage vector controller - reset time**					
Reset time for voltage vector control					
Setting range (min. value | unit | max. value)			Lenze setting		
0.01	ms	2000.00	3.77 ms		
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/100		UNSIGNED_32

5.14.3 Setting the voltage boost

As an alternative for the voltage vector control, a constant, load independent voltage boost can be specified for low speeds (below the V/f rated frequency) or for a motor standstill in order to optimise the starting performance.

Stop!

If the motor is operated at standstill for a longer time - especially in case of smaller motors - the motor can be destroyed by overtemperature!
- Connect the KTY of the motor. Parameterise and activate the Motor temperature monitoring. (📖 250)
- Parameterise and activate the Monitoring of the motor utilisation (I²xt). (📖 241)
- Operate self-ventilated motors with a blower, if required.

Note!

The voltage boost acts additively to the voltage vector control.
- Only use one of the two "Boost" functions.
  Recommendation: voltage vector control

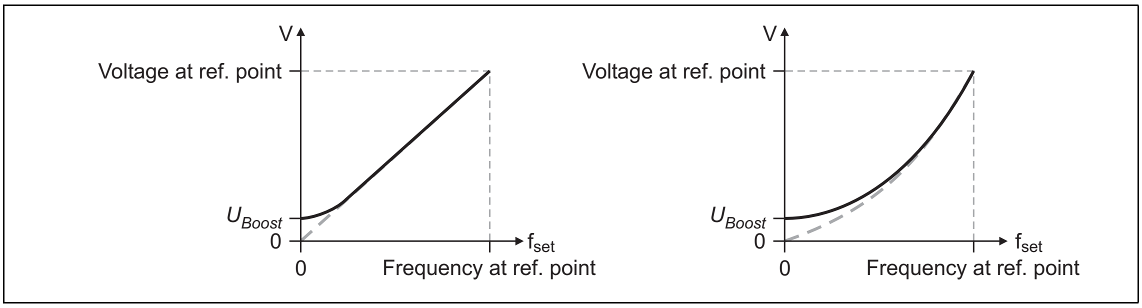
Depending on the required starting torque, the voltage boost must be set so that the required motor current will be available after controller enable.

- The voltage boost can be calculated by multiplying the stator resistance by the rated magnetising current:

$$\text{Starting current} \sim V_{Boost} = R_S \cdot I_{mN}$$

- Optionally, the voltage boost can be determined empirically by increasing the setting until the rated magnetising current flows.
- The voltage boost is added geometrically to the voltage of the characteristic:

$$U = \sqrt{U^2_{Characteristic} + U^2_{Boost}}$$



[5-11] Voltage boost for a linear V/f characteristic (left) and square-law V/f characteristic (right)

Tip!

For magnetising the motor, consider a sufficient time from the controller enable to the start of the speed ramp function generator.

- The bigger the motor the longer the time required for magnetisation. A motor with a power of 90 kW requires up to 2 seconds.

0x2B06 | 0x3306 - VFC: Voltage boost

Voltage boost

Setting range (min. value | unit | max. value)			Lenze setting	
0.0	V	100.0	0.0 V	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	UNSIGNED_16

5.14.4 Setting the load adjustment

Stop!

If the load adjustment is too high, the motor current may increase in idle state and the motor may overheat!

0x2B07 | 0x3307 - VFC: Load adjustment parameter

Sub.	Name	Lenze setting	Data type
1	VFC: Load adjustment - direction of rotation	0: Passive load (e.g. friction)	UNSIGNED_8
2	VFC: Load adjustment - value	20.00 %	UNSIGNED_32

Subindex 1: **VFC: Load adjustment - direction of rotation**			
Adaptation of the characteristic depending on the load in the case of CW and CCW rotation			
Selection list (Lenze setting printed in bold)		Info	
0	**Passive load (e.g. friction)**	The motor operates in motor mode in both directions.	
1	Active load, CCW (e.g. hoist)	Application example: Hoist without counterweight	
2	Active load, CW (e.g. hoist)	Application example: Dancer-controlled unwinder	
☑ Write access ☑ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_8

Subindex 2: **VFC: Load adjustment - value**					
Load adjustment in [%] proportionally to the rated torque, in order to obtain a correspondingly "rigid" drive behaviour even after the starting action. • When starting torque = rated torque, a load adjustment of 50 % is suitable for most applications.					
Setting range (min. value | unit | max. value)			Lenze setting		
0.00	%	200.00	20.00 %		
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/100		UNSIGNED_32

5.14.5 Defining the behaviour at the current limit (Imax controller)

The maximum output current or the current limit is determined by the object "Max. current" (0x6073 or 0x6873 for axis B). In case of the V/f characteristic control, an Imax controller is implemented for complying with this limit. If the motor current exceeds the set maximum value, the Imax controller is activated.

- The Imax controller changes the field frequency so that the motor current does not exceed the current limit. In motor mode, the frequency is reduced and in generator mode it is increased.
- The gain and reset time of the Imax controller can be parameterised.
  - For the automatic calculation of these two parameters, the "VFC: Calculate Imax controller parameters" function is provided via object 0x2822 (or 0x3022 for axis B).

Optimising the Imax controller

The automatic calculation serves to determine starting parameters of the Imax controller which are sufficient for many applications. Thus, an optimisation is not required for most of the applications.

The parameters of the Imax controller have to be adapted if

- a power control is implemented with great moments of inertia.
  Recommendation:
  Step 1: Increase reset time in 0x2B08:2 (or 0x3308:2 for axis B)
  Step 2: Reduce gain in 0x2B08:1 (or 0x3308:1 for axis B)
- vibrations occur with V/f characteristic control during the operation of the Imax controller.
  Recommendation:
  Step 1: Increase reset time in 0x2B08:2 (or 0x3308:2 for axis B)
  Step 2: Reduce gain in 0x2B08:1 (or 0x3308:1 for axis B)
- overcurrent errors occur due to load impulses or too high acceleration/deceleration ramps.
  Recommendation:
  Step 1: Reduce reset time in 0x2B08:2 (or 0x3308:2 for axis B)
  Step 2: Increase gain in 0x2B08:1 (or 0x3308:1 for axis B)

If the connected mechanics and the conditions in the machine allow it, it may be helpful to determine the optimal parameters of the current controller by a practical acceleration test with a reduced maximum current (0x6073 or 0x6873 for axis B). Mass inertia and acceleration/deceleration considerably determine the requirements of the Imax control loop.

Recommendation: Execute the adjustment with the real mass inertia and optimise the parameters step by step with an increasing acceleration/deceleration. The oscilloscope function of the »PLC Designers«/»EASY motor starter« serves to record the following objects:

- VFC: Setpoint frequency (0x2B0B or 0x330B for axis B)
- Motor: Actual voltage (0x2D82 or 0x3582 for axis B)
- Motor: Actual current (0x2DD1:5 or 0x35D1:5 for axis B)
- Device: Max. current (0x6073 or 0x6873 for axis B)
- Device: Actual output frequency (0x2DDD or 0x35DD for axis B)

0x2B08 | 0x3308 - VFC: Imax controller - Parameter

Sub.	Name	Lenze setting	Data type
1	VFC: Imax controller - gain	0.001 Hz/A	UNSIGNED_32
2	VFC: Imax controller - reset time	100.0 ms	UNSIGNED_32

Subindex 1: **VFC: Imax controller - gain**				
Gain for Imax controller				
Setting range (min. value | unit | max. value)			Lenze setting	
0.000	Hz/A	1000.000	0.001 Hz/A	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/1000	UNSIGNED_32

Subindex 2: **VFC: Imax controller - reset time**				
Reset time for Imax controller				
Setting range (min. value | unit | max. value)			Lenze setting	
1.0	ms	2000.0	100.0 ms	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	UNSIGNED_32

5.14.6 Setting the slip compensation

The slip compensation serves to automatically compensate a load-dependent speed loss. In order that the slip compensation can operate correctly, the rated slip of the motor is required. It is calculated from the rated motor frequency and the rated motor speed, thus both parameters must be parameterised correctly.

0x2B09 | 0x3309 - VFC: Slip compensation - Parameter

Sub.	Name	Lenze setting	Data type
1	VFC: Slip compensation - influence	0.00 %	INTEGER_16
2	VFC: Slip compensation - filter time	2000 ms	UNSIGNED_16

Subindex 1: **VFC: Slip compensation - influence**				
Adjustment in percent of the slip calculated • For instance required for deviations of the real motor data from the nameplate data. • A setting of 100 % corresponds to the rated slip of the machine.				
Setting range (min. value | unit | max. value)			Lenze setting	
-200.00	%	200.00	0.00 %	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/100	INTEGER_16

Subindex 2: **VFC: Slip compensation - filter time**				
Adjustment of the time-dependent behaviour of the slip compensation				
Setting range (min. value | unit | max. value)			Lenze setting	
1	ms	6000	2000 ms	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

5.14.7 Setting the oscillation damping

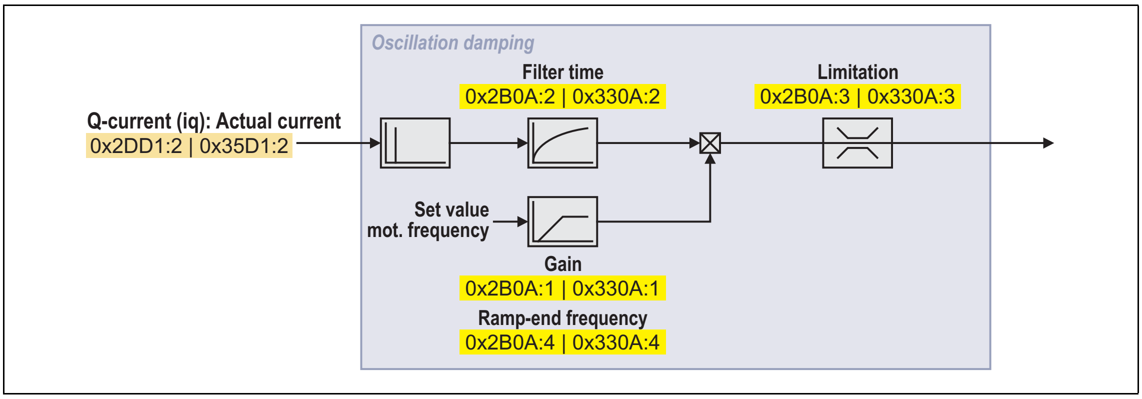
The oscillation damping serves to reduce the oscillations during no-load operation which are caused by energy oscillating between the mechanical system (mass inertia) and the electrical system (DC bus). Furthermore, the oscillation damping can also be used to compensate resonances.

Note!

Observe the following restrictions:
- Damping is possible only for constant oscillations at a steady-state operating point.
- Oscillations occurring sporadically cannot be damped.
- Oscillation damping is not suitable for oscillations occurring during dynamic processes (e.g. accelerations or load changes).
- Oscillation damping is only active if the setpoint speed is greater than 10 rpm and the DC-bus voltage exceeds a value of 100 V.

Function

The determination of the oscillation is based on the active current. In order to obtain the alternating component of the active current, this current is differentiated. This signal is then passed through a PT1 filter.



Identification of the oscillation

Before the oscillation damping can be parameterised, the oscillation has to be identified. One way to do this is to examine the motor current while oscillation damping is switched off (gain = 0 %). At steady-state operation, a constant current flows. If the drive oscillates, these oscillations are also visible on the motor current. It is therefore possible to determine the frequency and the amplitude of the oscillation from the alternating component of the motor current. In the following, this alternating component is referred to as "current oscillation".

Parameter setting

The gain of the oscillation signal is to be set according to the following equation:

$$\text{Gain of the oscillation signal} = \frac{\text{Current amplitude}}{\sqrt{2} \cdot \text{Maximum device current}} \cdot 100\ \%$$

The time constant of the PT1 filter has to be set in such a way that the oscillation can be damped and higher-frequency components are filtered out of the signal. The time constant is determined from the reciprocal value of the double frequency of the current oscillation.

$$\text{Time constant} = \frac{1}{2 \cdot \text{Oscillation frequency}}$$

The oscillation frequency calculated can be limited before it is added to the field frequency. The maximum frequency can be derived from the amplitude of the current oscillation, the rated motor current, and the slip frequency of the connected motor:

$$\text{Max. frequency} = \frac{2 \cdot \text{Amplitude of the current oscillation}}{\text{Rated motor current}} \cdot \text{Rated slip frequency}$$

0x2B0A | 0x330A - VFC: Oscillation damping - Parameter

Sub.	Name	Lenze setting	Data type
1	VFC: Oscillation damping - gain	20 %	INTEGER_16
2	VFC: Oscillation damping - filter time	5 ms	UNSIGNED_16
3	VFC: Oscillation damping - limitation	0.2 Hz	UNSIGNED_16
4	VFC: Oscillation damping - ramp-end frequency	0 %	UNSIGNED_8

Subindex 1: **VFC: Oscillation damping - gain**			
Gain of the oscillation signal • If the setting is "0 %", the oscillation damping is switched off.			
Setting range (min. value | unit | max. value)			Lenze setting
-100	%	100	20 %
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			INTEGER_16

Subindex 2: **VFC: Oscillation damping - filter time**			
Time constant of the PT1 filter			
Setting range (min. value | unit | max. value)			Lenze setting
1	ms	600	5 ms
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_16

Subindex 3: **VFC: Oscillation damping - limitation**			
Limitation of the calculated oscillation frequency before it is added to the field frequency.			
Setting range (min. value | unit | max. value)			Lenze setting
0.1	Hz	20.0	0.2 Hz
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX		Scaling: 1/10	UNSIGNED_16

Subindex 4: **VFC: Oscillation damping - final ramp frequency**				
Ramp end frequency from which the gain factor is to have reached its rated value. • By setting a ramp end frequency, a possible negative impact of the oscillation damping on the concentricity factor in the lower speed range can be reduced. • The ramp end frequency refers to the rated motor frequency in percent.				
Setting range (min. value | unit | max. value)			Lenze setting	
0	%	100	0 %	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_8

5.14.8 Optimising pull-out slip limitation

The following object serves to adapt the pull-out slip function or the maximally permissible current in the field weakening range.

- If the motor is to be stalled in the field weakening range, the override point can be offset by reducing the value so that a motor stalling can be prevented.
- If the motor cannot provide enough torque in the field weakening range, the value has to be increased.

0x2B0C | 0x330C - VFC: Override point of field weakening

From version 01.03
Offset of the override point for field weakening

Setting range (min. value | unit | max. value)			Lenze setting	
-500.0	Hz	500.0	0.0 Hz	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	INTEGER_16

5.14.9 Display parameter

0x2B0B | 0x330B - VFC: Setpoint frequency

Display area (min. value | unit | max. value)			Initialisation	
-600.0	Hz	600.0	0.0 Hz	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/10	INTEGER_16

5.14.10 "Flying restart" function

As a protective function against high compensation currents, the i700 servo inverter provides the flying restart function. High compensation currents may occur in the case of the V/f characteristic control if the drive is not at standstill at the time of controller enable. The flying restart function determines the motor speed by means of a test current and, by this information, presets the frequency setpoint. The speed determined is also provided to the axis control.

Stop!

If the flying restart function is deactivated and the controller is not enabled at standstill, the output voltage and output frequency do not match the current motor speed. High compensation currents may flow!

- The drive is first braked towards 0 Hz and is then accelerated again!

0x2BA0 | 0x33A0 - Flying restart: Activate

Activation of the additional "flying restart process" function

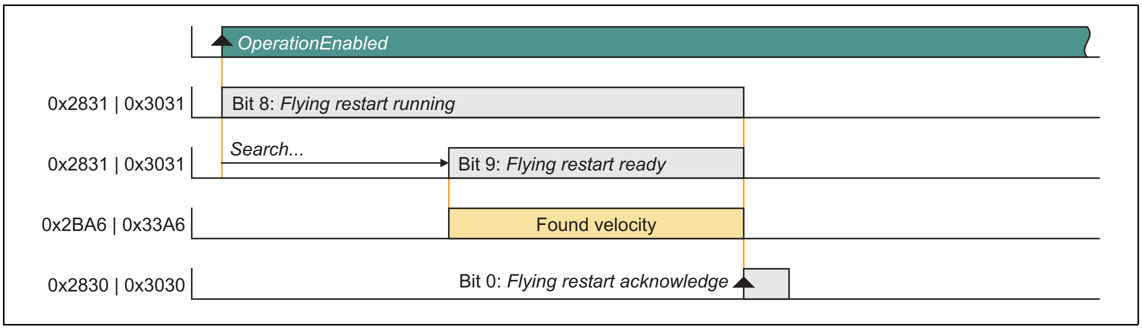
- If the flying restart function is activated ("1: On"), a flying restart process for the determination of the current motor speed is started automatically after the controller inhibit has been deactivated if the following conditions are met:
  - The V/f characteristic control is set as motor control.
  - The CiA402 mode is selected as drive mode.
  - The flying restart function is not blocked via bit 2 in the Lenze control word (0x2830 or 0x3030 for axis B).
  - No DC-injection braking is active.
  - No motor phase failure has been detected.

Selection list (Lenze setting printed in bold)		
0	**Off**	
1	On	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX		UNSIGNED_8

Flying restart process

If the "flying restart process" function is activated, the flying restart process will start after controller enable:

1. The i700 servo inverter reports the started flying restart process to the controller via bit 8 in the Lenze status word (0x2831 or 0x3031 for axis B).
2. If a speed has been found, it is reported to the controller via bit 9 in the Lenze-status word.
3. The controller gives the i700 servo inverter a feedback on the fact that the found speed has been accepted via bit 0 in the Lenze control word (0x2830 or 0x3030 for axis B). As long as the feedback is not provided, further flying restart processes are not possible.



[5-12] Signals during a flying restart process

Parameter setting

Note!

- The flying restart algorithm requires the motor voltage as exact as possible. Therefore it is absolutely necessary to predetermine the inverter error characteristic.
  ▶ Compensating for inverter influence on output voltage
- In addition to the exact motor voltage, the exact stator resistance must also be known. If the flying restart process does not work as desired, the setting of the stator resistance is to be slightly adapted in object 0x2C01:2 (or 0x3401:2 for axis B).
- A flying restart process can be blocked via bit 1 in the Lenze control word (0x2830 or 0x3030 for axis B).

A control loop is involved in the flying restart process the controller parameter integration time of which has to be adapted to the motor in the object 0x2BA3 (or 0x33A3 for axis B). For the automatic calculation of this parameter, the "VFC: Flying restart process controller - calculate controller parameters" function is available via the object 0x2822 (or 0x3022 for axis B).

The actual flying restart process can be adjusted via the following parameters:

0x2BA1 | 0x33A1 - Flying restart: Current

Current injected in the motor by the flying restart algorithm to identify the current speed.
- Selection in [%] based on the rated motor current (0x6075 or 0x6875 for axis B).
- The higher the current the higher the torque imposed upon the motor.
- If the current is too low, a wrong speed can be detected.

Setting range (min. value | unit | max. value)			Lenze setting	
0	%	100	15 %	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

0x2BA2 | 0x33A2 - Flying restart: Start frequency

Start frequency of the flying restart algorithm
- If it can be anticipated at which frequency the motor "caught" most frequently in a flying restart process, this frequency is to be set here.

Setting range (min. value | unit | max. value)			Lenze setting	
-600.0	Hz	600.0	20.0 Hz	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	INTEGER_16

0x2BA3 | 0x33A3 - Flying restart: Integration time

Integration time of the angle controller
- The Lenze setting is adapted to machines with an average power.
- A guide value for the integration time can be calculated as a function of the motor power with the following equation: Ti = 1.1 µ/W * rated motor power + 9.4 ms
- For accelerating the search process, this guide value can be reduced.
- If the flying restart frequency oscillates too much, the integration time has to be increased again.
- A longer integration time increases the time for "catching" the drive.

Setting range (min. value | unit | max. value)			Lenze setting	
1	ms	60000	600 ms	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

0x2BA4 | 0x33A4 - Flying restart: Min. deviation

Setting range (min. value | unit | max. value)			Lenze setting	
0.00	°	90.00	5.00 °	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/100	UNSIGNED_16

0x2BA5 | 0x33A5 - Flying restart: Delay time

To avoid starting a flying restart process at short-time controller inhibit, a time can be set here for the minimum active controller inhibit time before a flying restart process is activated.
- Since a pulse inhibit > 500 ms causes a controller inhibit, this also applies to the pulse inhibit.

Setting range (min. value | unit | max. value)			Lenze setting	
0	ms	10000	0 ms	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

0x2BA6 | 0x33A6 - Flying restart: Result

Output of the speed found for acceptance in the control.
- This information is pending until the control reports acceptance of the found speed to the i700 servo inverter via bit 0 in the Lenze control word (0x2830 or 0x3030 for axis B).

Sub.	Name	Lenze setting	Data type
1	Flying restart: Determined speed [rpm]		INTEGER_16
2	Flying restart: Determined speed [n unit]		INTEGER_32

Subindex 1: **Flying restart: Determined speed [rpm]**				
Display area (min. value | unit | max. value)			Initialisation	
-32768	r/min	32767		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX				INTEGER_16

Subindex 2: **Flying restart: Determined speed [n unit]**				
Display area (min. value | unit | max. value)			Initialisation	
-480000	[n unit]	480000		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: $480000/2^{31}$	INTEGER_32

5.15 "DC-injection braking" function

The control modes for asynchronous motors provide the opportunity to use "DC-injection braking" for the braking process.

DC-injection braking can be

- parameterised via bit 6 in the Lenze control word (0x2830 or 0x3030 for axis B) or
- as response to light errors. ▶ Response of the device in the event of an error (📖 262)

Cases of application

- for DC-injection braking via bit 6:
  - The braking energy cannot be converted into heat in a brake resistor.
  - An external control wants to request DC-injection braking with a drive in V/f operation.
- for DC-injection braking as response to light errors:
  - Due to an encoder error, no quick stop is possible anymore.

Functional description

In this case, the motor control injects a DC current the amplitude of which can be set in the object 0x2B80 (or 0x3380 for axis B). For this purpose, it is required that the current control is adapted to the corresponding motor. ▶ Setting and optimising the current controller (📖 105)

0x2B80 | 0x3380 - DC-injection braking: Current

Current amplitude for DC-injection braking
Note!
During the parameter setting of this object, it must be considered that a DC current amplitude is set here. In contrast, the rated current of the motor is usually given as effective value. In order to prevent an overload of the motor, the value set here should be at least lower by the factor "√2" than the rated motor current or attention should be paid on a correspondingly set I^2xt characteristic (0x2D4D or 0x354D for axis B).

Setting range (min. value | unit | max. value)			Lenze setting	
0.00	A	500.00	0.00 A	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/100	UNSIGNED_16

Note!

The r.m.s. value of the current is calculated from the time values by means of the use of the factor "√2". This also happens in case of the field frequency 0 Hz (DC current). In case of the DC injection braking, the r.m.s. value is thus given by the factor "√ too low in 0x2DD1:5 (or 0x35D1:5 for axis B).

5.16 "Short-circuit braking" function

The control modes for synchronous motors provide the opportunity to use "short-circuit braking" for the braking process.

Short-circuit braking can be

- parameterised via bit 6 in the Lenze control word (0x2830 or 0x3030 for axis B) or
- as response to light errors. ▶ Response of the device in the event of an error (📖 262)

> **Stop!**
>
> In some constellations, it is not possible to decelerate the motor speed of a synchronous motor to zero by means of "short-circuit braking". The braking effect cannot be compared to the "quick stop" function where the motor is energised under optimal conditions!

Cases of application

- for short-circuit braking via bit 6:
  - The braking energy cannot be converted into heat in a brake resistor.
  - An external control wants to request short-circuit braking as e.g. an error has been detected in the encoder system which does not permit braking via quick stop.
- for short-circuit braking as response to light errors:
  - Due to an encoder error, no quick stop is possible anymore.

Functional description

> **Note!**
>
> The short-circuit current is set freely according to the motor voltage (k_E * speed) and the internal resistance of the system. Thus, it is mandatory that the ampacity of the i700 servo inverter is oriented towards the short-circuit current to be maximally expected.
>
> Guide value: I_{max_device} (3 s) ≥ 1.5 * I_{max_motor} (according to the data sheet/catalogue)
>
> In case of a deviating assignment, a rating based on the currently possible parameters (max. speed, max. motor current, field weakening, etc.) is required!

The effect of the short-circuit braking on the deceleration behaviour depends on the motor features, the effective cable length, the mass inertia and the initial value of the speed (application point).

Primarily, short-circuit braking can be used to convert a part of the kinetic energy into heat energy which serves to unload external brake assemblies and limit position dampers.

If short-circuit braking is to be used as the only holistically effective deceleration unit, it is recommended to verify the feasibility by tests. For this purpose, bit 6 in the Lenze control word (0x2830 or 0x3030 for axis B) serves to trigger the short-circuit braking. The oscilloscope function of the »PLC Designer«/»EASY Starter« serves to record the following important objects:

- Speed: Setpoint speed (0x2DD3:1 or 0x35D3:1 for axis B)
- Speed: Actual speed (0x606C or 0x686C for axis B)
- Phase currents U, V, W (0x2D83:2...4 or 0x3583:2...4 for axis B)

5.17 Setting the switching frequency

The i700 servo inverter can be operated with the switching frequencies listed below.

0x2939 | 0x3139 - Switching frequency

Note: In case of "8 kHz variable" and "16 kHz variable", variable switching frequencies are involved. If the permissible continuous current for the respective switching frequency is exceeded, it is automatically switched back to the next-lower switching frequency and vice versa.
For servo applications, we recommend, if possible, the operation with a fix switching frequency!

Selection list (Lenze setting printed in bold)		Info
0	16 kHz variable	• Low operating noises of the motor but higher current consumption during the acceleration phases. • For operating motors which are designed for switching frequencies > 8 kHz, e.g. milling drives and mid-frequency motors.
1	8 kHz variable	• Reduced operating noises of the motor but higher current consumption during the acceleration phases compared to the operation with 4 kHz. • E.g. for the operation of internally-ventilated asynchronous servo motors.
2	**4 kHz**	• Servo applications with medium requirements or control quality. • V/f characteristic control • Operation of motors which are designed for switching frequencies of 3 to 6 kHz.
10	16 kHz fix	From version 01.06 • Servo application with highest requirements or control quality. • Operation of motors which are designed for switching frequencies of 12 to 16 kHz, e.g. mid-frequency motors. • Low-noise operation of motors.
11	8 kHz fix	From version 01.06 • Servo application with high requirements or control quality. • Servo control SM with field weakening • Operation of motors which are designed for switching frequencies of 6 to 10 kHz.
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX		UNSIGNED_8

5.18 Frequency and speed limitations

Output frequency

The output frequency of the i700 servo inverter is limited to a maximum value whose amount corresponds to the lower of the two following values:

$$f_{lim} = \frac{f_{chop}}{8} \quad \text{or} \quad f_{lim} = f_{max\_device}$$

f_{lim}	Maximum output frequency
f_{chop}	Switching frequency (0x2939 or 0x3139 for axis B)
f_{max_device}	Maximum device output frequency (currently 1999 Hz; dependent on the device version)

Speed setpoint

If servo control is used, the speed setpoint is limited depending on the number of motor pole pairs:

$$n_{lim} = \frac{f_{lim}}{zp}$$

n_{lim}	Speed limit value
f_{lim}	Maximum output frequency
zp	Number of motor pole pairs

- If the speed setpoint is limited, bit 1 ("Speed: Setpoint 1 limited") or bit 5 ("Speed: Setpoint 2 limited") is set in the Lenze status word (0x2831 or 0x3031 for axis B).
- The behaviour corresponds to the behaviour which is shown when the set maximum speed (0x6080 or 0x6880 for axis B) is reached.
- The sequence is: Limit the speed to 0x6080 (or 0x6880 for axis B) first, then limit it to the speed limit value n_{lim}.

Frequency setpoint

If V/f characteristic control is used, the frequency setpoint is limited in addition to the speed setpoint.

- If the frequency setpoint is limited, bit 10 ("Output frequency limited") is set in the Lenze status word (0x2831 or 0x3031 for axis B).

6 Holding brake control

This device function is used to control a motor holding brake connected to the i700 servo inverter.

Note!

In the "Fault" device status, the holding brake is applied.

Objects described in this chapter

Object		Name	Data type
Axis A	Axis B		
0x2820	0x3020	Brake control: Settings	RECORD

6.1 Operating modes

For holding brake control, the following three operating modes available for the i700 servo inverter in the object 0x2820:1 (or 0x3020:1 for axis B):

- Triggering via control word via external control (Lenze setting)
- Triggering via state machine of device
- No brake connected

Tip!

We recommend the use of the preset "Triggering via control word from external control" mode. In this operating mode, the triggering is carried out via the application program in the Controller.

You can find detailed information on the respective operating mode in the following subchapters.

6.1.1 Triggering via control word via external control (Lenze setting)

Object		Name	Required setting
Axis A	Axis B		
0x2820:1	0x3020:1	Brake: Operating mode	1: Activation via control word by ext. control system

In this preset operating mode, the triggering takes place via the application program in the Controller via bit 14 in the CiA402 control word (0x6040 or 0x6840 for axis B).

- Bit 14 in the CiA402 control word = "0" → close holding brake
- Bit 14 in the CiA402 control word = "1" → release holding brake

Function block L_SMC_BrakeControl

For Lenze Controllers, the **L_SMC_BrakeControl** function block is available for control. It contains the following functions:

- Torque feedforward control in case of release in order to prevent a sagging for active loads (hoists).
- Considering the delay times of the holding brake during the opening and closing process.
  - The delay times which have been transferred as parameters to the function block from the application are considered.
  - The brake parameters "closing time" and "opening time" of the i700 servo inverter are not effective in this operating mode.
- Option to manually release the holding brake for service purposes via a control input.

The basic conditions and timing are displayed in the illustration "Principal signal flow of the holding brake control".

6.1.2 Triggering via state machine of device

Object		Name	Required setting
Axis A	Axis B		
0x2820:1	0x3020:1	Brake: Operating mode	0: Triggering via device state machine

In this operating mode, the holding brake is triggered as a function of the device status. ▶ Device states (📖 179)

Note!

In the event of an error or when STO ("SafeTorqueOff") is activated, closing of the brake and inhibit of the operation take place immediately without considering the set brake closing time.

The basic conditions and timing are displayed in the illustration "Principal signal flow of the holding brake control".

6.1.3 No brake connected

Object		Name	Required setting
Axis A	Axis B		
0x2820:1	0x3020:1	Brake: Operating mode	2: No brake connected

In this operating mode, no control, detection and monitoring of the brake takes place during normal operation.

6.2 Display of the holding brake status

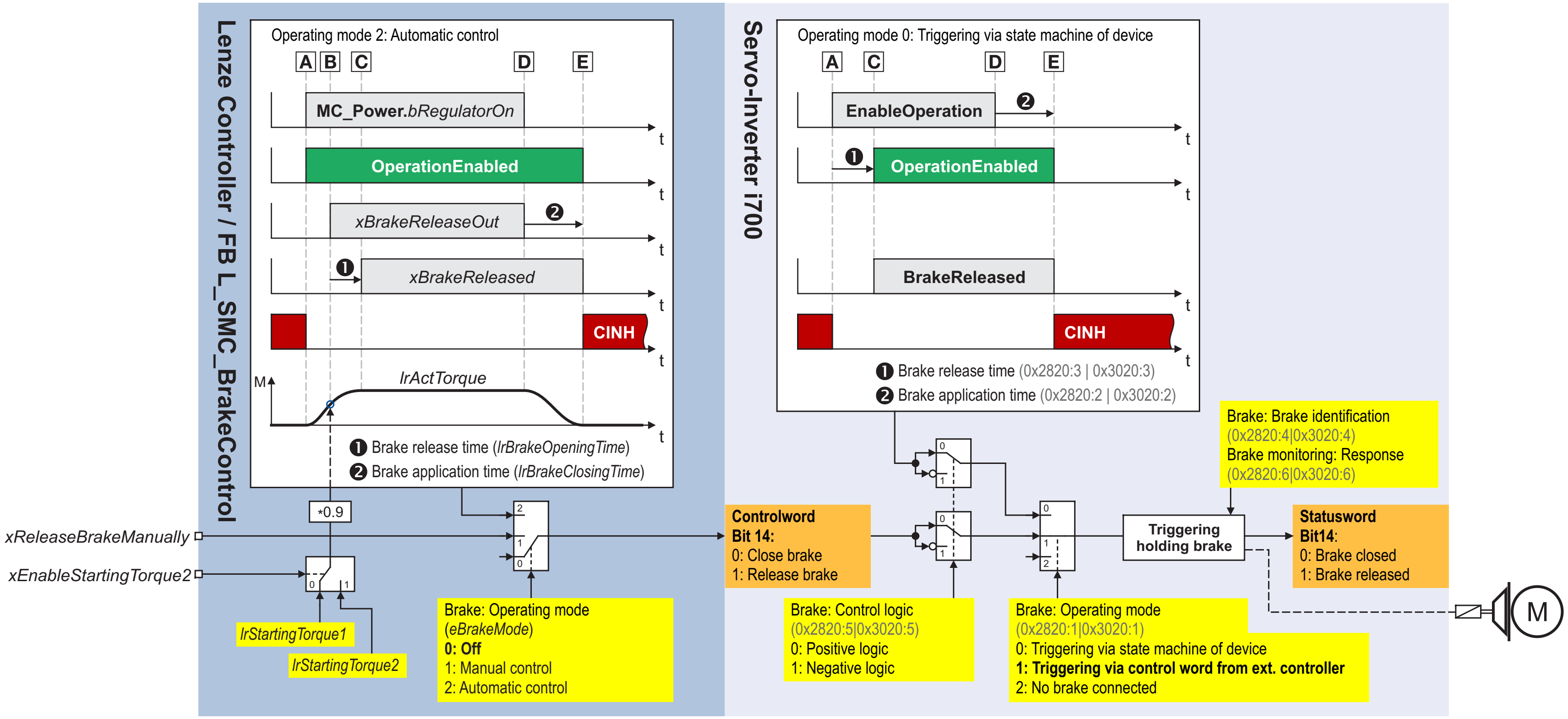
The CiA402 status word (0x6041 or 0x6841 for axis B) displays the holding brake status via bit 14:

- Bit 14 in the CiA402 status word = "0" means "Holding brake applied"
- Bit 14 in the CiA402 status word = "1" means "Holding brake released"

6.3 Basic signal flow

The following signal flow clarifies the interaction between the **L_SMC_BrakeControl** function block (in the application program of the Lenze Controller) and the device function integrated in the Servo Inverter i700 for holding brake control.

Principal signal flow of the holding brake control (with the respective operating mode compared to the automatic control)



L_SMC_BrakeControl - process of the automatic control:	
A	When **MC_Power**.*bRegulatorOn* = TRUE, the i700 is enabled and the starting torque for the torque feedforward control is specified simultaneously.
B	When the actual torque is = 0.9 x starting torque, the holding brake is triggered to be released. The triggering is signalled at the *xBrakeReleaseOut* output.
C	After the brake opening time has elapsed, the "holding brake released" status is reported via the *xBrakeReleased* output. ➜ From now on, the Servo Inverter i700 can perform motions.
D	When **MC_Power**.*bRegulatorOn* = FALSE, the holding brake is triggered to be closed.
E	Only after the brake closing time has elapsed, the i700 is inhibited. The *xBrakeReleased* output reports the "Holding brake closed" status.

Servo Inverter i700 - process of the control via device status machine:	
A	The "Enable operation" command serves to enable the i700 and simultaneously trigger the holding brake to be released.
C	After the brake opening time has elapsed, the "holding brake released" status is reported via the CiA402 status word (bit 14). ➜ From now on, the Servo Inverter i700 can perform motions.
D	The "Disable operation" command serves to trigger the holding brake to be closed.
E	Only after the brake closing time has elapsed, the i700 is inhibited. The CiA402 status word (bit 14) serves to report the "Holding brake closed" status.

6.4 Settings

0x2820 | 0x3020 - brake control: settings

Sub.	Name	Lenze setting	Data type
1	Brake: Operating mode	1: Activation via control word by ext. control system	UNSIGNED_8
2	Brake: Application time	100 ms	UNSIGNED_16
3	Brake: Release time	100 ms	UNSIGNED_16
4	Brake: Brake identification	0: Identification not active	UNSIGNED_16
5	Brake: Control logic	0: Positive logic	UNSIGNED_8
6	Brake monitoring: Response	1: Trouble	UNSIGNED_8

Subindex 1: **Brake: operating mode**		
Note: In the case of the manual control (0x2825 = "4" or 0x3025 = "4" for axis B), the holding brake is controlled automatically. The setting of the operating mode is irrelevant in this case.		
Selection list (Lenze setting printed in bold)		Info
0	Triggering via state machine of device	The holding brake is triggered as a function of the device status. ▸ More information
1	**Triggering via control word from external controller**	The holding brake is triggered via the application program in the Controller. ▸ More information
2	No brake connected	During normal operation, no control, detection, and monitoring of the brakes take place.
☑ Write access ☐ CINH ☐ OSC ☑ P ☑ RX ☐ TX		UNSIGNED_8

Subindex 2: **Brake: application time**			
Only effective in the "Triggering via device state machine" operating mode.			
Setting range (min. value | unit | max. value)			Lenze setting
0	ms	10000	100 ms
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_16

Subindex 3: **Brake: release time**			
Only effective in the "Triggering via device state machine" operating mode.			
Setting range (min. value | unit | max. value)			Lenze setting
0	ms	10000	100 ms
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_16

Subindex 4: **Brake: brake identification**		
In the case of the manual control (0x2825 = "4" or 0x3025 = "4" for axis B), during the changeover from the "Switched on" to "Operation enabled" device status, it is automatically determined whether a holding brake is connected. The brake identification is executed again after every controller enable.		
Selection list (read only)		
0	**Detection not active**	
1	Brake detection is running...	
2	No brake detected	
3	Brake detected	
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX		UNSIGNED_16

Subindex 5: **Brake: Control logic**			
The control logic of the holding brake can be inverted.			
Selection list (Lenze setting printed in bold)		Info	
0	**Positive logic**	Positive logic means: • Release of the holding brake: The 1-signal serves to trigger the brake output via bit 14 in the CiA402 control word or by the device state machine (relay contact closed). • In the CiA402 status word, bit 14 is set (holding brake released). Negative logic means: • Level inversion of the relay output. • Release of the holding brake: The 1-signal does not trigger the brake output via bit 14 in the CiA402 control word or by the device state machine (relay contact is open). • In the CiA402 status word, bit 14 is set (holding brake released).	
1	Negative logic		
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_8

Subindex 6: **Brake monitoring: Response**			
In the controlled state, the holding brake is monitored cyclically for the presence of the brake current. Since, after connecting the holding brake, the brake current builds up in a time-delayed manner as a function of inductance, an open circuit, a terminal short circuit, or a missing brake supply is detected with a delay. When monitoring trips, the response set here will be activated. **Note:** The holding brake is not monitored in the non-triggered state!			
Selection list (Lenze setting printed in bold)			
0	No response		
1	**Fault**		
2	Warning		
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_8

7 CiA402 device profile

The CiA402 device profile describes predefined functions or modes of a variable-speed drive. The individual process data for the modes are clearly defined. The axis parameterisation and the bus configuration (PDO mapping) are carried out via standardised indexes.

Objects described in this chapter

Object		Name	Data type
Axis A	Axis B		
0x2500		Touch probe (TP): Filter time	UNSIGNED_16
0x2946	0x3146	Cyclic sync torque mode: Speed limitation	RECORD
0x2D00	0x3500	Touch probe (TP): Delay time	RECORD
0x2D01	0x3501	Touch probe (TP): Time stamp	RECORD
0x6040	0x6840	Controlword	UNSIGNED_16
0x6041	0x6841	Statusword	UNSIGNED_16
0x6042	0x6842	vl target velocity	INTEGER_16
0x6043	0x6843	vl velocity demand	INTEGER_16
0x6044	0x6844	vl velocity actual value	INTEGER_16
0x6046	0x6846	vl velocity min max amount	RECORD
0x6048	0x6848	vl velocity acceleration	RECORD
0x6049	0x6849	vl velocity deceleration	RECORD
0x605A	0x685A	Quick stop option code	INTEGER_16
0x6060	0x6860	Operating mode: Selection	INTEGER_8
0x6061	0x6861	Modes of operation display	INTEGER_8
0x6062	0x6862	Position demand value	INTEGER_32
0x6063	0x6863	Position actual internal value	INTEGER_32
0x6064	0x6864	Position actual value	INTEGER_32
0x6065	0x6865	Following error window	UNSIGNED_32
0x6066	0x6866	Following error time out	UNSIGNED_16
0x6067	0x6867	Position window	UNSIGNED_32
0x6068	0x6868	Position window time	UNSIGNED_16
0x606C	0x686C	Velocity actual value	INTEGER_32
0x6071	0x6871	Target torque	INTEGER_16
0x6072	0x6872	Max torque	UNSIGNED_16
0x6073	0x6873	Max current	UNSIGNED_16
0x6074	0x6874	Target torque	INTEGER_16
0x6077	0x6877	Torque actual value	INTEGER_16
0x6078	0x6878	Current actual value	INTEGER_16
0x6079	0x6879	DC bus: Actual voltage	UNSIGNED_32
0x607A	0x687A	Position demand value	INTEGER_32
0x607E	0x687E	Polarity	UNSIGNED_8
0x6080	0x6880	Max motor speed	UNSIGNED_32
0x6085	0x6885	Quick stop deceleration	UNSIGNED_32
0x608F	0x688F	Position encoder resolution	RECORD
0x6090	0x6890	Velocity encoder resolution	RECORD

Object		Name	Data type
Axis A	Axis B		
0x60B1	0x68B1	Velocity offset	INTEGER_32
0x60B2	0x68B2	Torque offset	INTEGER_16
0x60B8	0x68B8	Touch probe function	UNSIGNED_16
0x60B9	0x68B9	Touch probe status	UNSIGNED_16
0x60BA	0x68BA	Touch probe pos1 pos value	INTEGER_32
0x60BB	0x68BB	Touch probe pos1 neg value	INTEGER_32
0x60BC	0x68BC	Touch probe pos2 pos value	INTEGER_32
0x60BD	0x68BD	Touch probe pos2 neg value	INTEGER_32
0x60C0	0x68C0	Interpolation sub mode select	INTEGER_16
0x60C2	0x68C2	Interpolation time period	RECORD
0x60E0	0x68E0	Positive torque limit value	UNSIGNED_16
0x60E1	0x68E1	Negative torque limit value	UNSIGNED_16
0x60F4	0x68F4	Following error actual value	INTEGER_32
0x60FA	0x68FA	Control effort	INTEGER_32
0x60FC	0x68FC	Position demand internal value	INTEGER_32
0x60FD	0x68FD	Digital inputs	UNSIGNED_32
0x60FF	0x68FF	Target velocity	INTEGER_32
0x6404	0x6C04	Motor manufacturer	STRING(50)
0x6502	0x6D02	Supported drive modes	UNSIGNED_32
0x67FF	0x6FFF	ECAT: Device Profile Number	UNSIGNED_32

CiA402 objects described in other chapters:

Object		Name	Data type
Axis A	Axis B		
▸ Indication of fault and warning (error code)			
0x603F	0x683F	Error code	UNSIGNED_16
0x605E	0x685E	Fault reaction option code	INTEGER_16
▸ Set motor parameters manually			
0x6075	0x6875	Motor rated current	UNSIGNED_32
0x6076	0x6876	Motor rated torque	UNSIGNED_32

7.1 Supported drive modes

CiA402 - drive modes	Can be used with	
	Servo control	V/f characteristic control
Velocity mode (vl) • Speed control	●	●
Cyclic sync velocity mode (csv) • Speed control with interpolation of the speed setpoint.	●	●
Cyclic sync torque mode (cst) • Torque control with interpolation of the torque setpoint.	●	
Cyclic sync position mode (csp) • Position control with interpolation of the position setpoint.	●	

Setpoint interpolation in the operating modes with a cyclic setpoint selection

When an operating mode with cyclic setpoint selection is selected, first all setpoints are controlled via interpolators, which divide the setpoint step-changes down from the bus cycle to the cycle time of the control loops. All interpolators are parameterised commonly via the "Interpolation time period" object (0x60C2 or 0x68C2 for axis B).

7.2 Applied units and scaling for position and velocity

Name	Unit	Scaling
CiA402 position	Pos unit	2^{16} ... 2^{30} [Pos unit] ≡ 360° Depending on the position encoder resolution set (0x608F or 0x688F for axis B).
Internal position	Incr.	2^{32} [Incr.] ≡ 360°
CiA402 velocity	r/min	
Internal velocity	n-unit	$\pm 2^{31}$ [n-unit] ≡ ±480000 r/min

7.3 General CiA402 parameters

This chapter describes the general CiA402 parameters.

All objects correspond to the CiA402 specification, but some of them have only a restricted value range.

Objects described in this chapter:

Object Axis A	Object Axis B	Name	Data type
0x60FD	0x68FD	Digital inputs	UNSIGNED_32
0x6404	0x6C04	Motor manufacturer	STRING(50)
0x6502	0x6D02	Supported drive modes	UNSIGNED_32
0x67FF	0x6FFF	ECAT: Device Profile Number	UNSIGNED_32
Greyed out = read access only			

Objects described in other chapters:

Object Axis A	Object Axis B	Name	Data type
0x603F	0x683F	Error code	UNSIGNED_16
Greyed out = read access only			

0x60FD | 0x68FD - Digital inputs

Display of the current status of the digital inputs

Display area (min. value | unit | max. value)			Initialisation
0		4294967295	0
Value is bit-coded:			Info
Bits 0-3	Not specified		
Bits 4-15	Reserved		
Bit 16	Level at digital input 1		0: LOW level 1: HIGH level
Bit 17	Level at digital input 2		0: LOW level 1: HIGH level
Bits 18-31	Reserved		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			UNSIGNED_32

Related topics:
- Touch probe (TP)
- Delay times of the digital inputs and required minimum signal lengths

0x6404 | 0x6C04 - Motor manufacturer

☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX		STRING(50)

0x6502 | 0x6D02 - Supported drive modes

Bit coded display of the operating modes in which the drive can be actuated.
0: Operating mode is not supported.
1: Operating mode is supported.

Display area (min. value | unit | max. value)			Initialisation	
0x00000000		0xFFFFFFFF	0x00000382	
Value is bit-coded:			Info	
Bit 0	Not specified		Not supported	
Bit 1	Velocity mode		▶ Velocity mode	
Bit 2	Not specified		Not supported	
Bit 3	Not specified		Not supported	
Bit 4	Reserved			
Bit 5	Not specified		Not supported	
Bit 6	Not specified		Not supported	
Bit 7	Cyclic sync position mode		▶ Cyclic sync position mode	
Bit 8	Cyclic sync velocity mode		▶ Cyclic sync velocity mode	
Bit 9	Cyclic sync torque mode		▶ Cyclic sync torque mode	
Bits 10-15	Reserved			
Bits 16-31	Reserved			
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX				UNSIGNED_32

0x67FF | 0x6FFF - Device profile number

Display area (min. value | unit | max. value)			Initialisation	
0x00000000		0xFFFFFFFF	0x00020192	
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX				UNSIGNED_32

7.4 Device control

The objects described in this chapter serve to control the states of the controller and select the operating mode.

All objects correspond to the CiA402 specification, but some of them have only a restricted value range.

Objects described in this chapter:

Object Axis A	 Axis B	Name	Data type
0x6040	0x6840	Controlword	UNSIGNED_16
0x6041	0x6841	Statusword	UNSIGNED_16
0x605A	0x685A	Quick stop option code	INTEGER_16
0x6060	0x6860	Operating mode: Selection	INTEGER_8
0x6061	0x6861	Modes of operation display	INTEGER_8
0x6085	0x6885	Quick stop deceleration	UNSIGNED_32
Greyed out = read access only			

0x6040 | 0x6840 - CiA402 controlword

Control word for the drive

Setting range (min. value | unit | max. value)			Lenze setting
0		65535	0
Value is bit-coded: (☑ = bit set)			Info
Bit 0 ☐	Switch on		
Bit 1 ☐	DC bus: Establish readiness for operation		
Bit 2 ☐	Activate quick stop		
Bit 3 ☐	Enable operation		
Bits 4-6 ☐	Depending on the operating mode		
Bit 7 ☐	Fault reset		▶ Indication of fault and warning (error code)
Bit 8 ☐	Not specified		
Bit 9 ☐	Depending on the operating mode		
Bit 10 ☐	Reserved		
Bit 11 ☐	Reserved		
Bit 12 ☐	Reserved		
Bit 13 ☐	Reserved		
Bit 14 ☐	Release brake		▶ Holding brake control
Bit 15 ☐	Reserved		
☑ Write access ☐ CINH ☑ OSC ☐ P ☑ RX ☐ TX			UNSIGNED_16

0x6041 | 0x6841 - Statusword

Status word from the drive

Display area (min. value | unit | max. value)			Initialisation
0		65535	
Value is bit-coded:			Info
Bit 0	Ready to switch on		
Bit 1	Switched on		
Bit 2	Operation enabled		
Bit 3	Trouble active		
Bit 4	Voltage enabled		
Bit 5	Quick stop		
Bit 6	Switch on disabled		
Bit 7	Warning is active		
Bit 8	Deactivate RPDOs		Cyclic PDOs have been deactivated in 0x2824 (or 0x3024 for axis B).
Bit 9	Remote		Drive can receive commands via EtherCAT.
Bit 10	Target reched		The actual position is located in the window (0x6067 or 0x6867 for axis B).
Bit 11	Internal limitation is active		For details see Lenze status word (0x2831 or 0x3031 for axis B).
Bit 12	Drive Follows Command		Operation is enabled and no test mode is activated (no internal setpoint generation active).
Bit 13	Following error		The position could not be approached.
Bit 14	Brake released		
Bit 15	STO is not active		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			UNSIGNED_16

0x605A | 0x685A - Quick stop option code

Device status after exiting the quick stop ramp
2 = automatic changeover to the "Switch-on disabled" device status
6 = the axis remains in the "Quick stop active" device status.

Setting range (min. value | unit | max. value)			Lenze setting
2		6	2
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			INTEGER_16

0x6060 | 0x6860 - Modes of operation

Selection of the operating mode:
0 = no operating mode (standstill)
2 = Velocity mode
8 = Cyclic sync position mode
9 = Cyclic sync velocity mode
10 = Cyclic sync torque mode

Setting range (min. value | unit | max. value)			Lenze setting
0		10	0
☑ Write access ☐ CINH ☑ OSC ☑ P ☑ RX ☐ TX			INTEGER_8

0x6061 | 0x6861 - Modes of operation display

Display of the current operating mode:
0 = no operating mode (standstill)
2 = Velocity mode
8 = Cyclic sync position mode
9 = Cyclic sync velocity mode
10 = Cyclic sync torque mode

Display area (min. value | unit | max. value)			Initialisation
0		10	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			INTEGER_8

0x6085 | 0x6885 - Quick stop deceleration

Change in velocity used for deceleration to a standstill if quick stop is activated.
- In consideration of the position encoder resolution set in 0x608F (or 0x688F for axis B) (see sample calculation below).

Setting range (min. value | unit | max. value)			Lenze setting
0	[Pos unit/s²]	2147483647	2147483647 [Pos unit/s²]
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_32

Example: Calculation of the deceleration to be set at a given ramp time

Specifications:
- Position encoder resolution (0x608F) = "16 bits/revolution".
  ➜ A mechanical motor revolution is indicated with 65536 increments.
- Initial speed of the motor = 1000 rpm
- Duration of the ramp until standstill = 2.5 s

Deceleration to be set:

$$\text{Delay} = \frac{\text{Initial speed}}{\text{Period}} \cdot \frac{1}{60} \cdot \frac{\text{0x608F:1}}{\text{0x608F:2}}$$

$$\text{0x6085} = \frac{1000\ [\text{r/min}]}{2.5\ [\text{s}]} \cdot \frac{1}{60\ [\text{s/min}]} \cdot \frac{65536\ [\text{increments}]}{1\ [\text{r}]}$$

$$\text{0x6085} = 436907 \left[\frac{\text{Increments}}{s^2}\right]$$

7.4.1 Commands for the device status control

Via the Controlword (0x6040 or 0x6840 for axis B) commands can be executed to make the controller change to a specific device status:

Command	Bit pattern in the Controlword							
	Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Fault reset	Control bits depending on the operating mode			Enable operation	Activate quick stop	Enable voltage	Switch on
Shutdown	0	X	X	X	X	1	1	0
Switch on	0	X	X	X	X	1	1	1
Enable operation	0	X	X	X	1	1	1	1
Activate quick stop	0	X	X	X	X	0	1	X
Disable operation	0	X	X	X	0	1	1	1
Disable voltage	0	X	X	X	X	X	0	X
Fault reset	0 ↗ 1	X	X	X	X	X	X	X
X = Status not significant								

Tip!

The greyed out control bits listed in the table are not important for the activation of commands but only serve to improve readability of the bit patterns.

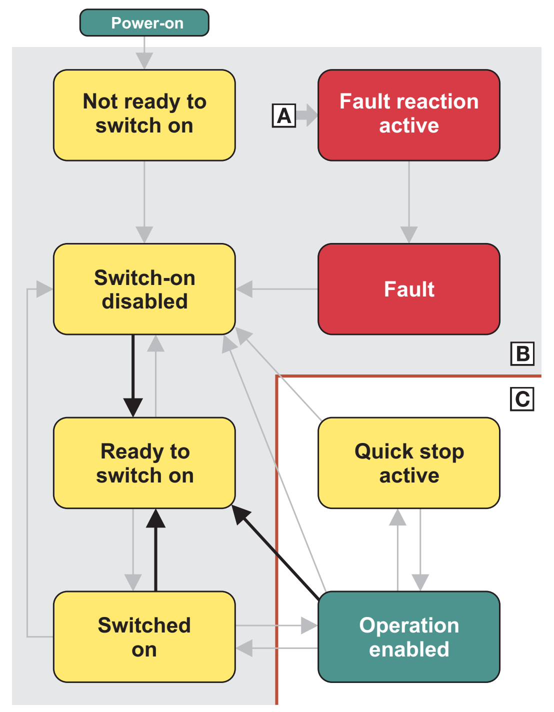
A PLC program of a PLCopen control can, for instance, trigger several commands for state changes in a row by the level change at the *bRegulatorOn* input of the "MC_Power" block. In the mentioned example, these device commands are "switch-off" and "switch-on" in this order.

Detailed information on the different commands can be obtained from the following subchapters.

Further Lenze-specific control bits (bits 8 ... 15)

Device control command	Bit pattern in the Controlword							
	Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
	Reserved	Release brake	Reserved					
Apply brake	X	0	X	X	X	X	X	X
Release brake	X	1	X	X	X	X	X	X
X = Status not significant								

7.4.1.1 Shutdown



A From all states
B Power section inhibited (pulse inhibit)
C Power section enabled

This command changes the device status from "Switch on disabled" to "Ready to switch on".

If the pulse inhibit has already been deactivated and the device status of the controller is "Operation enabled", this command can be used to set the pulse inhibit again.

- If automatic brake operation is activated, the parameterised brake closing time is observed: The system waits until the brake is applied before the pulse inhibit is set.
- The motor becomes torqueless.
- The device status "Switched on" or "Operation enabled" changes back to "Ready to switch on".

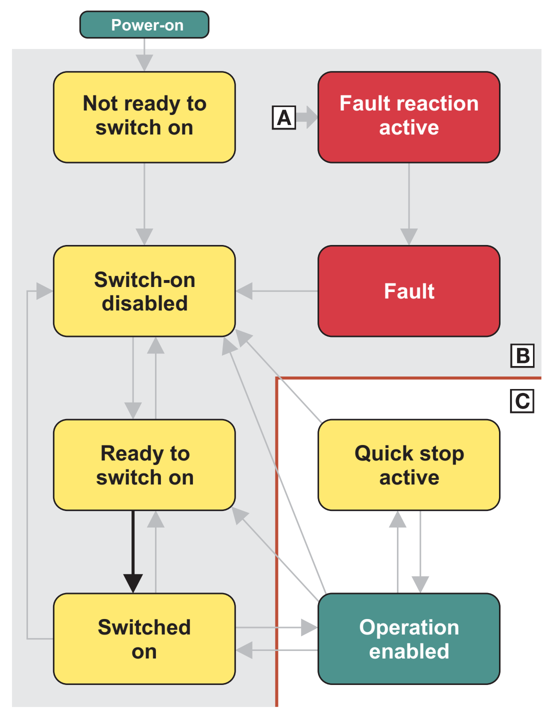
Danger!

If the motor gets torqueless and a load is connected, motors without a holding brake may move in an uncontrolled way!
In case no load is connected, the motor is coasting.

Bit pattern for the "Shutdown" command in the Controlword (0x6040 or 0x6840 for axis B):

Bits 15 - 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Fault reset	Control bits depending on the operating mode			Enable operation	Activate quick stop	Enable voltage	Switch on
X	0	X	X	X	X	1	1	0

7.4.1.2 Switch on



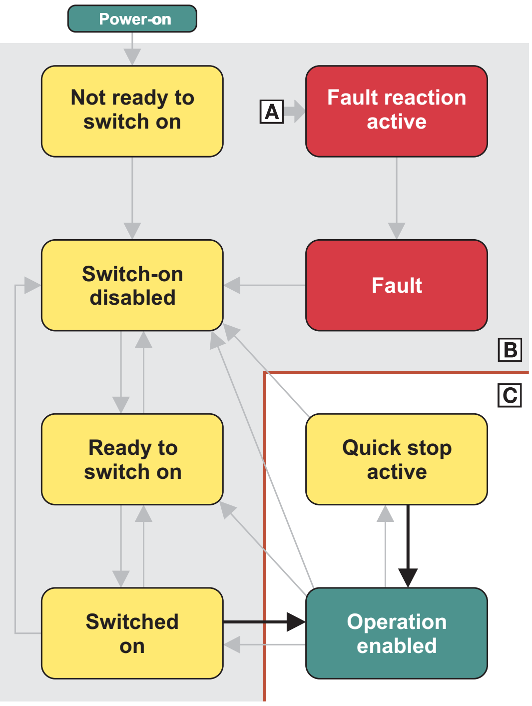
A From all states
B Power section inhibited (pulse inhibit)
C Power section enabled

This command serves to deactivate the switch on inhibit which is active after switch on or after the reset (acknowledgement) of an error.

- A changeover to the "Switched on" device status takes place.

Bit pattern for the "Switch on" command in the Controlword (0x6040 or 0x6840 for axis B):								
Bits 15 - 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Fault reset	Control bits depending on the operating mode			Enable operation	Activate quick stop	Enable voltage	Switch on
X	0	X	X	X	X	1	1	1

7.4.1.3 Enable operation



A From all states
B Power section inhibited (pulse inhibit)
C Power section enabled

This command serves to enable the operation and stop an active quick stop again.

- A changeover to the "Operation enabled" device status takes place.
- The output stages of the controller become active.

Bit pattern for the "Enable operation" command in the Controlword (0x6040 or 0x6840 for axis B):								
Bits 15 - 8	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Fault reset	Control bits depending on the operating mode			Enable operation	Activate quick stop	Enable voltage	Switch on
X	0	X	X	X	1	1	1	1

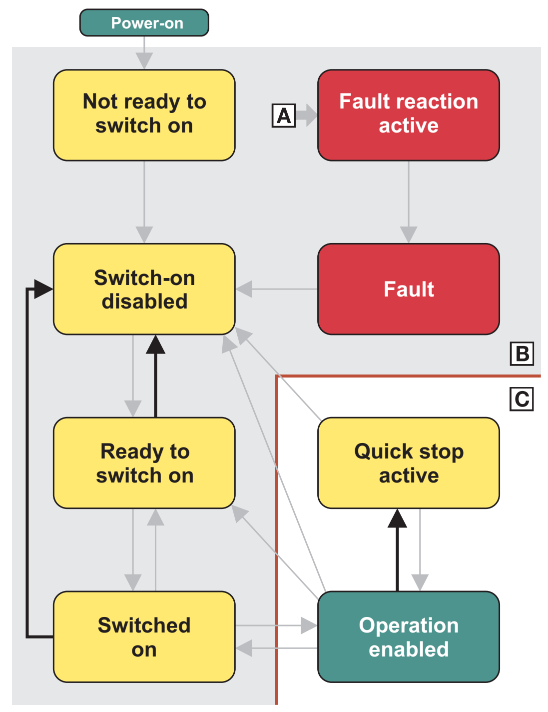
Note!

The signalling of the "Operation enabled" device status in the CiA402 status word can be delayed in the following cases:

- If in case of the synchronous motor servo control the "pole position identification option has been activated before the start in 0x2C63 (or 0x3463 for axis B) and is just running (few milliseconds).
- If the brake is in the "control via device state machine" mode and the brake opening time (0x2820 or 0x3020 for axis B) has not elapsed yet.
- If an asynchronous motor is used which has not been magnetised yet.
  Check the→ setting of the rated motor current (0x6075 or 0x6875 for axis B) and the maximum device current (0x6073 or 0x6873 for axis B).

Only when the "Operation enabled" device status is signalled in the CiA402 status word, the points mentioned before are concluded and the i700 servo inverter is ready for the acceptance of setpoints of the Controller.

7.4.1.4 Activate quick stop



A From all states
B Power section inhibited (pulse inhibit)
C Power section enabled

This command serves to activate quick stop when operation is enabled.

- Irrespective of the setpoint specified, the drive is brought to a standstill with the deceleration set for quick stop (0x6085 or 0x6885 for axis B).
- A changeover to the "Quick stop is active" device status takes place.

If the operation has not been enabled yet (status "Ready to switch on" or "Switched on"), this command triggers a change to the "Switch on disabled" device status.

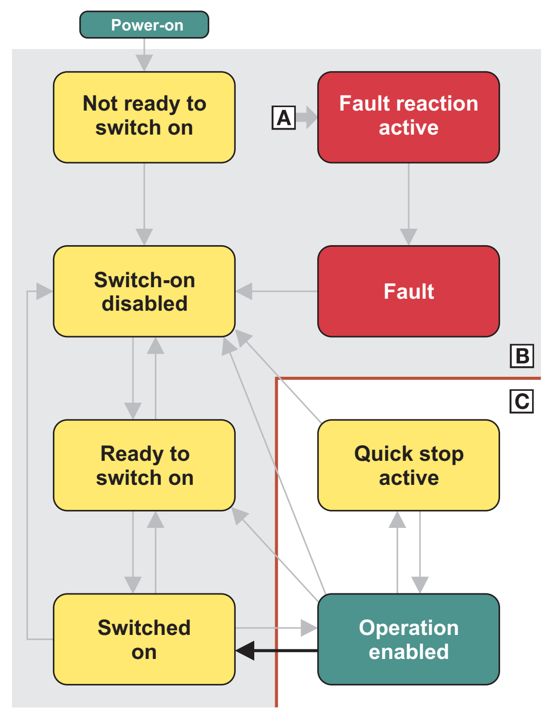
Bit pattern for the "Quick stop" in the Controlword (0x6040 or 0x6840 for axis B):								
Bits 15 - 8	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Fault reset	Control bits depending on the operating mode			Enable operation	Activate quick stop	Enable voltage	Switch on
X	0	X	X	X	X	0	1	X

Note!

During quick stop, the drive executes the setpoint generation and no longer follows the setpoint defined by the controller!

If several drives execute a chained synchronous motion, the quick stop function has to be coordinated by the Controller by means of a quick stop profile (master function). In this case, quick stop cannot be activated via the control bit 6!

7.4.1.5 Disable operation



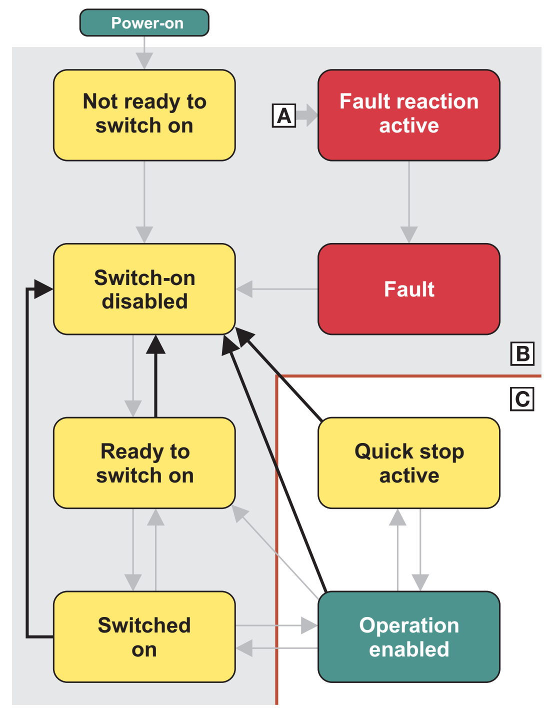
A From all states
B Power section inhibited (pulse inhibit)
C Power section enabled

This command serves to inhibit the enabled operation again.

- The pulse inhibit is set.
- If automatic brake operation is activated, the parameterised brake closing time is observed: The system waits until the brake is applied before the pulse inhibit is set.
- A changeover back to the "Switched on" device status takes place.

Bit pattern for the "Disable operation" command in the Controlword (0x6040 or 0x6840 for axis B):								
Bits 15 - 8	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Fault reset	Control bits depending on the operating mode			Enable operation	Activate quick stop	Enable voltage	Switch on
X	0	X	X	X	0	1	1	1

7.4.1.6 Disable voltage



A From all states
B Power section inhibited (pulse inhibit)
C Power section enabled

This command serves to inhibit the output stages of the controller again.

- The pulse inhibit is set (unless they are set already), i.e. the pulses of the controller are inhibited.
- The motor becomes torqueless.
- A changeover back to the "Switch on disabled" device status takes place.

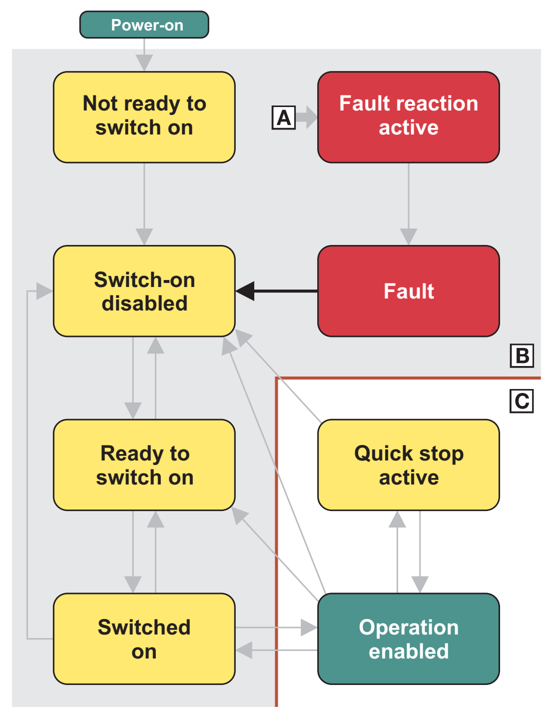
Danger!

If the motor gets torqueless and a load is connected, motors without a holding brake may move in an uncontrolled way!
In case no load is connected, the motor is coasting.

Bit pattern for the "Disable voltage" in the Controlword (0x6040 or 0x6840 for axis B):

Bits 15 - 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Fault reset	Control bits depending on the operating mode			Enable operation	Activate quick stop	Enable voltage	Switch on
X	0	X	X	X	X	X	0	X

7.4.1.7 Fault reset



[A] From all states
[B] Power section inhibited (pulse inhibit)
[C] Power section enabled

This command serves to reset an existing fault if the cause of the fault has been eliminated.

- The pulse inhibit remains set.
- A changeover to the "Switch on disabled" device status takes place, i. e. the switch on inhibit is active.

Bit pattern for the "Reset fault" in the Controlword (0x6040 or 0x6840 for axis B):								
Bits 15 - 8	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Fault reset	Control bits depending on the operating mode			Enable operation	Activate quick stop	Enable voltage	Switch on
X	0 ↗ 1	X	X	X	X	X	X	X

7.4.2 Device states

The current device status of the drive can be read via the Statusword (0x6041 or 0x6841 for axis B):

Device status	Bit pattern in the Statusword							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Warning is active	Switch on disabled	Quick stop	Voltage enabled	Fault active	Operation enabled	Switched on	Ready to switch on
Not ready to switch on	X	0	X	X	0	0	0	0
Switch on disabled	X	1	X	X	0	0	0	0
Ready to switch on	X	0	1	X	0	0	0	1
Switched on	X	0	1	X	0	0	1	1
Operation enabled	X	0	1	X	0	1	1	1
Quick stop is active	X	0	0	X	0	1	1	1
Fault reaction active	X	0	X	X	1	1	1	1
Fault	X	0	X	X	1	0	0	0
X = Status not significant								

Tip!

Bits 4 ("Voltage enabled") and 7 ("Warning active") are not relevant as regards the device status and merely serve the purpose of a better readability of the bit patterns here.

Detailed information on the different device states can be obtained from the following subchapters.

"Warning active" status bit

Via bit 7 in the Statusword, a warning is indicated.

- The occurrence of a warning does not cause a state change.
- Warnings do not need to be reset.

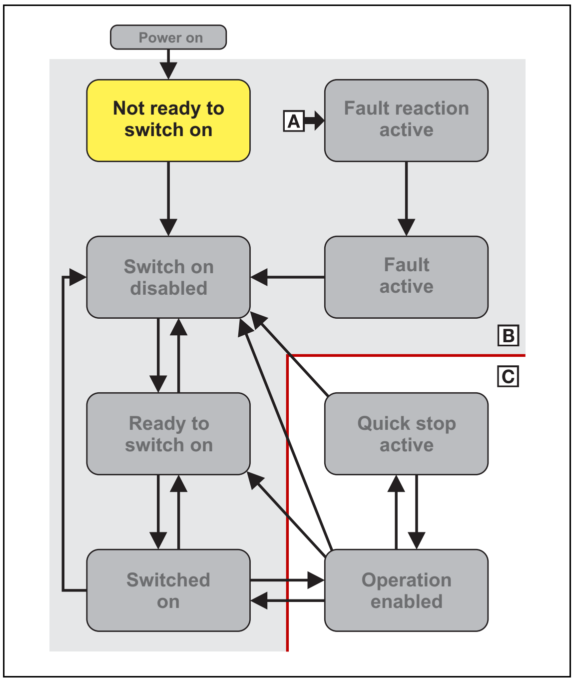
Further Lenze-specific status bits (bits 8 ... 15)

Device status	Bit pattern in the status word							
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	STO is not active	Brake released	Following error	Drive follows setpoint selection	Internal limitation is active	Target reched	Remote	RPDOs deactivated
Brake applied	X	0	X	X	X	X	X	X
Brake released	X	1	X	X	X	X	X	X
STO is active	0	X	X	X	X	X	X	X
STO is not active	1	X	X	X	X	X	X	X
X = Status not significant								

Note!

If STO is inhibited, a changeover to the "Switch on disabled" device status takes place and a warning is transmitted.

7.4.2.1 Not ready to switch on



This is the status of the controller directly after switching on the supply voltage.

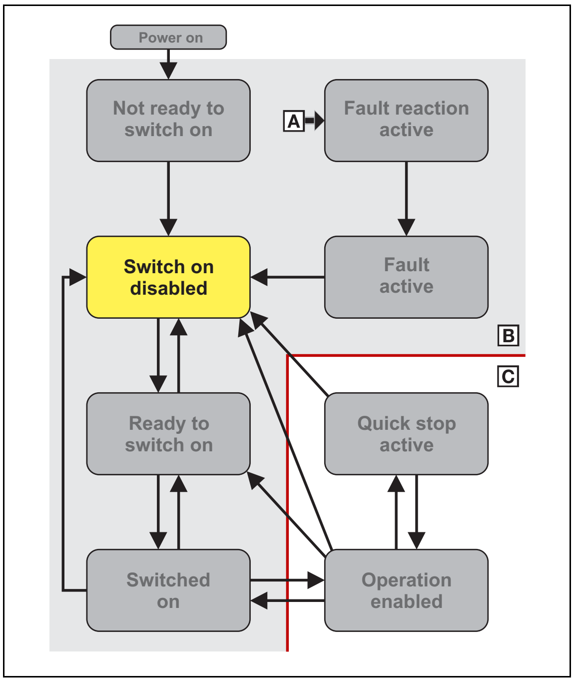
- In this device status, the device is initialised.
- Communication is not possible yet.
- The controller cannot be parameterised yet and no device commands can be carried out yet.
- The motor brake, if available, is closed.
- The operation is inhibited.

Bit pattern for the "Not ready to switch on" device status in the Statusword (0x6041 or 0x6841 for axis B):								
Bits 15 - 8	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Warning is active	Switch on disabled	Quick stop	Voltage enabled	Fault active	Operation enabled	Switched on	Ready to switch on
X	X	0	X	X	0	0	0	0

Note!

The controller changes to "Switch on disabled" if the EtherCAT bus is in the "Operational" status, or PDO communication is deactivated via object 0x2824 (or 0x3024 for axis B).

7.4.2.2 Switch on disabled

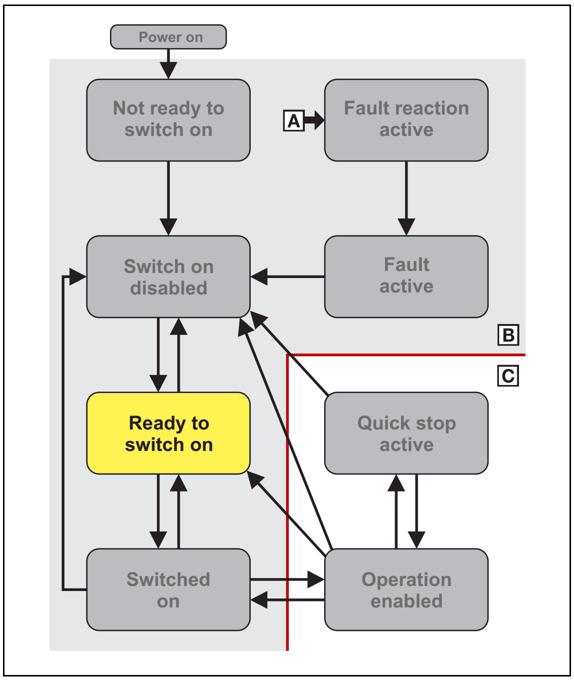


This is the controller's status after the device has been initialised successfully.

- The process data monitoring is active.
- Communication is possible.
- DC-bus voltage is available.
- The inverter can be parameterised.
- The motor brake, if available, is closed.
- The operation is inhibited.

Bit pattern for the "Switch on disabled" device status in the Statusword (0x6041 or 0x6841 for axis B):								
Bits 15 - 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Warning is active	Switch on disabled	Quick stop	Voltage enabled	Fault active	Operation enabled	Switched on	Ready to switch on
X	X	1	X	X	0	0	0	0

7.4.2.3 Ready to switch on



This is the device status of the controller after the device has been initialised successfully and after the controller has received the "Shutdown" command.

- The process data monitoring is active.
- Communication is possible.
- DC-bus voltage is available.
- The inverter can be parameterised.
- The motor brake, if available, is closed.
- The operation is inhibited.

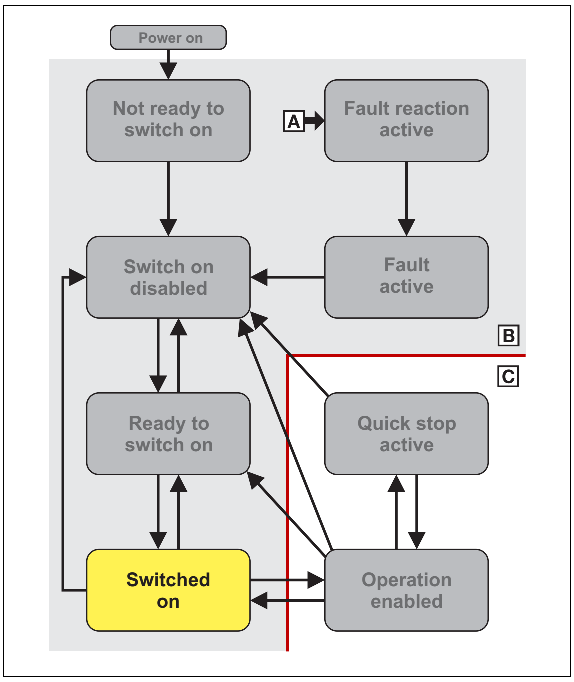
Bit pattern for the "Ready to switch on" device status in the Statusword (0x6041 or 0x6841 for axis B):								
Bits 15 - 8	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Warning is active	Switch on disabled	Quick stop	Voltage enabled	Fault active	Operation enabled	Switched on	Ready to switch on
X	X	0	1	X	0	0	0	1

Note!

A changeover to this device status is also effected if, in the "Switched on" or "Operation enabled" device status, the "Shutdown" command is activated.

The change to the sequential "Switched on" state is effected by activation of the "Switch on" command.

7.4.2.4 Switched on



This is the status of the controller after, in the "Ready to switch on" device status, it has received the "Switch on".

- The process data monitoring is active.
- Communication is possible.
- DC-bus voltage is available.
- The inverter can be parameterised.
- The motor brake, if available, is closed.
- The operation is inhibited.

Bit pattern for the "Switched on" device status in the Statusword (0x6041 or 0x6841 for axis B):								
Bits 15 - 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Warning is active	Switch on disabled	Quick stop	Voltage enabled	Fault active	Operation enabled	Switched on	Ready to switch on
X	X	0	1	X	0	0	1	1

Note!

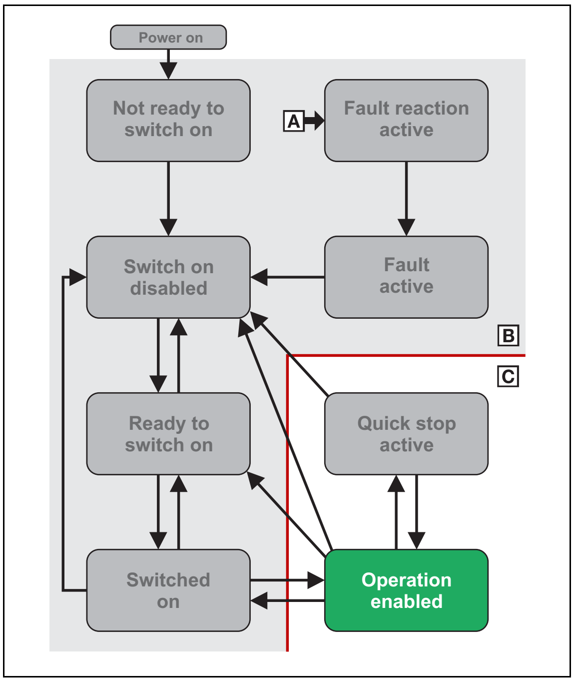
The changeover to the "Operation enabled" device status, and thus to operation enable, is effected by triggering the "Enable operation" command.

The signalling of the "Operation enabled" device status in the CiA402 status word can be delayed in the following cases:

- If in case of the synchronous motor servo control the "pole position identification option has been activated before the start in 0x2C63 (or 0x3463 for axis B) and is just running (few milliseconds).
- If the brake is in the "control via device state machine" mode and the brake opening time (0x2820 or 0x3020 for axis B) has not elapsed yet.
- If an asynchronous motor is used which has not been magnetised yet.
  Check the→ setting of the rated motor current (0x6075 or 0x6875 for axis B) and the maximum device current (0x6073 or 0x6873 for axis B).

Only when the "Operation enabled" device status is signalled in the CiA402 status word, the points mentioned before are concluded and the i700 servo inverter is ready for the acceptance of setpoints from the control.

7.4.2.5 Operation enabled

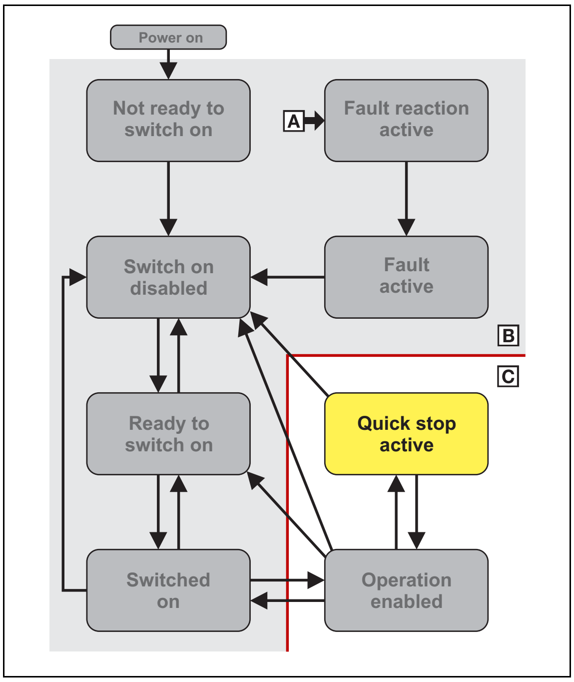


This device status represents normal operation. The operation in the operating mode selected is enabled and no errors are pending.

- Only the parameters of the inverter can be changed which do not require controller inhibit.
- A possibly available motor brake is released if the automatic operating mode of the Holding brake control is activated.
- The drive control is working.

Bit pattern for the "Operation enabled" device status in the Statusword (0x6041 or 0x6841 for axis B):								
Bits 15 - 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Warning is active	Switch on disabled	Quick stop	Voltage enabled	Fault active	Operation enabled	Switched on	Ready to switch on
X	X	0	1	X	0	1	1	1

7.4.2.6 Quick stop is active



This device status is active if quick stop is executed or is active.

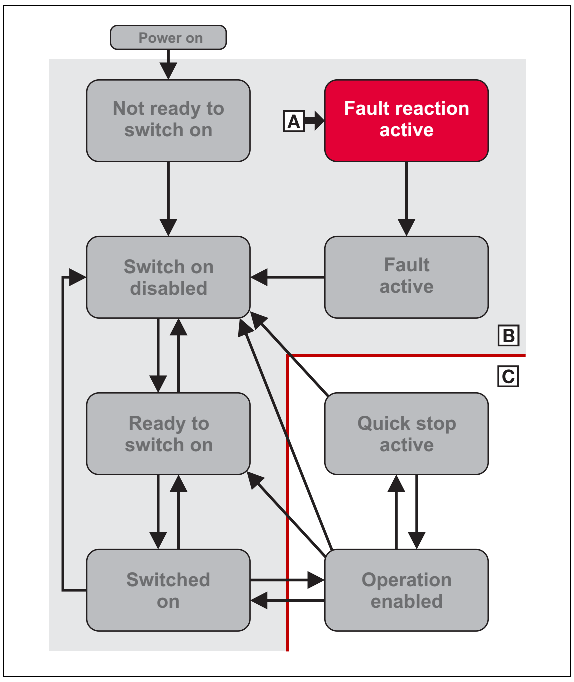
- Only the parameters of the inverter can be changed which do not require controller inhibit.
- The motor brake, if available, is opened.
- The drive control is working.

Bit pattern for the "Quick stop active" device status in the Statusword (0x6041 or 0x6841 for axis B):								
Bits 15 - 8	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Warning is active	Switch on disabled	Quick stop	Voltage enabled	Fault active	Operation enabled	Switched on	Ready to switch on
X	X	0	0	X	0	1	1	1

Note!

The "Enable operation" command serves to deactivate an active quick stop again.

7.4.2.7 Fault reaction active



If a minor fault occurs, i. e. the drive is still able to actuate the motor in a controlled manner, this device status becomes active and the drive is brought to a standstill irrespective of the setpoint specified with the deceleration set for quick stop (0x6085 or 0x6885 for axis B).

- Only the parameters of the inverter can be changed which do not require controller inhibit.
- The motor brake, if available, is opened.
- The drive control is working.

Bit pattern for the "Fault reaction active" device status in the Statusword (0x6041 or 0x6841 for axis B):

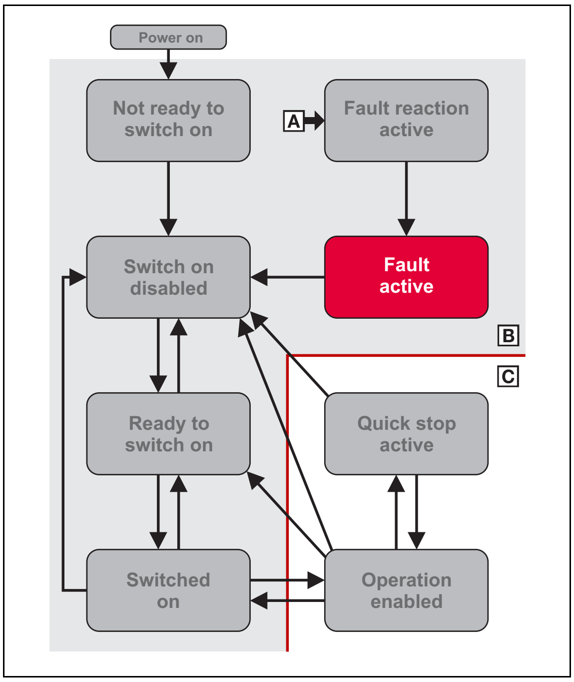
Bits 15 - 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Warning is active	Switch on disabled	Quick stop	Voltage enabled	Fault active	Operation enabled	Switched on	Ready to switch on
X	X	0	X	X	1	1	1	1

Note!

After quick stop has been executed, i. e. when the drive is at standstill, an automatic changeover to the "Fault" device status is effected.

The "Fault" device status can only be exited using the "Fault reset" command if the cause of the fault is eliminated.

7.4.2.8 Fault



If a grave error occurs, i.e. the drive is not able to operate the motor in a controlled manner anymore, the drive is switched off immediately and this device state becomes active.

- The pulse inhibit is set, i.e. the pulses of the controller are inhibited.
- The motor is torqueless.
- The motor brake, if available, is closed.
- The operation is inhibited.
- The inverter can be parameterised.

Bit pattern for the "Fault" device status in the Statusword (0x6041 or 0x6841 for axis B):

Bits 15 - 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Warning is active	Switch on disabled	Quick stop	Voltage enabled	Fault active	Operation enabled	Switched on	Ready to switch on
X	X	0	X	X	1	0	0	0

Note!

This device status can only be left using the "Fault reset" command if the cause of the fault is eliminated.

7.4.3 Selection of the operating mode

The drive behaviour depends on the selected operating mode. The operating mode is selected by the Controller via the 0x6060 object (or 0x6860 for axis B) in the communication statuses "Pre-Operational" or "Operational". Access can be effected via SDO or PDO.

- Only one operating mode at a time can be active.
- The operating modes differ by the type of setpoint selection.
- The following table shows the permissible operating modes with the corresponding object value:

Modes of operation	Object value	Setpoint selection
Velocity mode (vl)	2	Velocity
Cyclic sync position mode (csp)	8	Position
Cyclic sync velocity mode (csv)	9	Velocity
Cyclic sync torque mode (cst)	10	Torque
-	All other values	- (Standstill)

Tip!

The operating mode currently set can be read via the 0x6061 object (or 0x6861 for axis B). For this object, too, access is possible via SDO or PDO.

7.5 Parameters for the scaling of physical values

Objects described in this chapter:

Object Axis A	 Axis B	Name	Data type
0x607E	0x687E	Polarity	UNSIGNED_8
0x6080	0x6880	Max motor speed	UNSIGNED_32
0x608F	0x688F	Position encoder resolution	RECORD
0x6090	0x6890	Velocity encoder resolution	RECORD

0x607E | 0x687E - Polarity

Mounting position
- Only the setting "0" is accepted.

Setting range (min. value | unit | max. value)			Lenze setting	
0		0	0	
☑ Write access ☑ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_8

0x6080 | 0x6880 - Max. motor speed

Speed limitation

Setting range (min. value | unit | max. value)			Lenze setting	
0	r/min	480000	6075 r/min	
☑ Write access ☐ CINH ☑ OSC ☑ P ☑ RX ☐ TX				UNSIGNED_32

0x608F | 0x688F - Position encoder resolution

Resolution of the position detection by the motor encoder

Sub.	Name	Lenze setting	Data type
1	Encoder increments	0x00010000: 16 Bit	UNSIGNED_32
2	Motor revolutions	1	UNSIGNED_32

Subindex 1: **Encoder increments**			
Setting of the number of bits with which a mechanical motor revolution is to be resolved.			
Selection list (Lenze setting printed in bold)			
0x00010000	**16 bits**		
0x00040000	18 bits		
0x00100000	20 bits		
0x00400000	22 bits		
0x01000000	24 bits		
0x04000000	26 bits		
0x10000000	28 bits		
0x40000000	30 bits		
☑ Write access ☑ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_32

Subindex 2: **Motor revolutions**			
Number of motor revolutions • Only the setting "1" is accepted.			
Setting range (min. value | unit | max. value)			Lenze setting
1		1	1
☑ Write access ☑ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_32

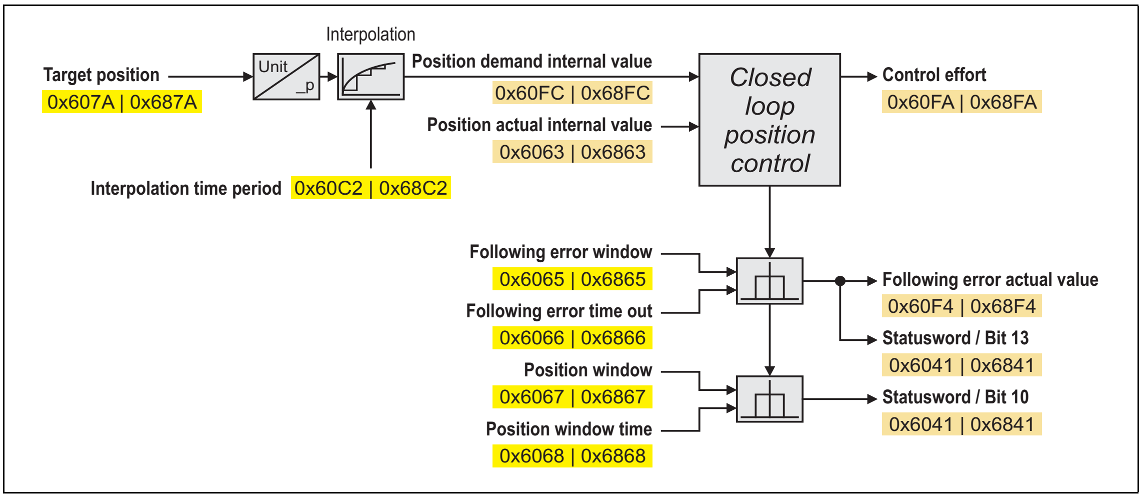
0x6090 | 0x6890 - Velocity encoder resolution

Note: This object is not used in the firmware and only available for compatibility reasons. Do not use the contained values for a calculation on the control level.

Sub.	Name	Lenze setting	Data type
1	Incr./s	33554432	UNSIGNED_32
2	Motor revolutions/s	125	UNSIGNED_32
☑ Write access ☑ CINH ☐ OSC ☑ P ☐ RX ☐ TX			

7.6 Parameters for actuation of the position control

The objects described in this chapter actuate the position control (including the following error and in-position recognition). All objects comply with the CiA402 specification.



[7-1] Access to the position control via the CiA402 device profile

Input data

Object		Name	Data type
Axis A	Axis B		
0x60FC	0x68FC	Position demand internal value	INTEGER_32
0x6062	0x6862	Position demand value	INTEGER_32
0x6065	0x6865	Following error window	UNSIGNED_32
0x6066	0x6866	Following error time out	UNSIGNED_16
0x6067	0x6867	Position window	UNSIGNED_32
0x6068	0x6868	Position window time	UNSIGNED_16
Greyed out = read access only			

Output data

Object		Name	Data type
Axis A	Axis B		
0x6063	0x6863	Position actual internal value	INTEGER_32
0x6064	0x6864	Position actual value	INTEGER_32
0x60F4	0x68F4	Following error actual value	INTEGER_32
0x60FA	0x68FA	Control effort	INTEGER_32
▶ Device control			
0x6041	0x6841	Statusword	UNSIGNED_16
Greyed out = read access only			

0x6062 | 0x6862 - Position demand value

Interpolated set position for the position control

Display area (min. value | unit | max. value)			Initialisation	
-2147483648	[Pos unit]	2147483647	0 [Pos unit]	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☐ TX				INTEGER_32

0x6063 | 0x6863 - Position actual internal value

Actual position in internal unit

Display area (min. value | unit | max. value)			Initialisation	
-2147483648	Incr.	2147483647		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX				INTEGER_32

0x6064 | 0x6864 - Position actual value

Current position

Display area (min. value | unit | max. value)			Initialisation	
-2147483648	[Pos unit]	2147483647		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX				INTEGER_32

0x6065 | 0x6865 - Following error window

Symmetrical tolerance window around the set position for the purpose of following error detection

- If the tolerance window is set = "0", the following error detection is deactivated.
- If the tolerance window is set > "0", the following error detection is activated, i.e. a following error will be detected if the actual position is outside this tolerance window.
- If the following error detection is active: If the following error is detected for a period longer than that defined in [ms] in object 0x6066 (or 0x6866 for axis B), bit 13 ("Following error") is set in the Statusword (0x6041 or 0x6841 for axis B).
- Object 0x60F4 (or 0x68F4 for axis B) shows the current deviation of the actual position to the set position.

Setting range (min. value | unit | max. value)			Lenze setting	
0	[Pos unit]	4294967295	1000 [Pos unit]	
☑ Write access ☐ CINH ☑ OSC ☑ P ☑ RX ☐ TX				UNSIGNED_32

0x6066 | 0x6866 - Following error time out

Time monitoring for following error detection
0 = following error is evaluated without time delay.

Setting range (min. value | unit | max. value)			Lenze setting	
0	ms	0	0 ms	
☑ Write access ☐ CINH ☑ OSC ☑ P ☑ RX ☐ TX				UNSIGNED_16

0x6067 | 0x6867 - Position window

Symmetrical tolerance window around the target position for in-position recognition
- If the actual position is within this tolerance window for a longer period than for the time defined in [ms] in object 0x6068 (or 0x6868 for axis B) , the target position is considered as reached, and bit 10 ("Target position reached") is set in the Statusword (0x6041 or 0x6841 for axis B).

Setting range (min. value | unit | max. value)			Lenze setting	
0	[Pos unit]	4294967295	1000 [Pos unit]	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_32

0x6068 | 0x6868 - Position window time

Time monitoring for the status message "target position reached"
0 = position in the target window is evaluated without time delay.

Setting range (min. value | unit | max. value)			Lenze setting	
0	ms	0	0 ms	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

0x60F4 | 0x68F4 - Following error actual value

Current following error

Display area (min. value | unit | max. value)			Initialisation	
-2147483648	[Pos unit]	2147483647		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX				INTEGER_32

0x60FA | 0x68FA - Control effort

Target position for the position control

Display area (min. value | unit | max. value)			Initialisation	
-480000	[n unit]	480000		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: $480000/2^{31}$	INTEGER_32

0x60FC | 0x68FC - Position demand internal value

Interpolated set position for the position control in the internal unit

Display area (min. value | unit | max. value)			Initialisation	
-2147483648	Incr.	2147483647	0 incr.	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☐ TX				INTEGER_32

7.7 Velocity mode (vl)

Selection of the operating mode

"Velocity mode" is selected with the setting "2" in 0x6060 (or 0x6860 for axis B).

7.7.1 Default mapping

The default mapping for the "Velocity mode" is defined in the following objects:

Object		Name	Data type
Axis A	Axis B		
0x1603	0x1613	RPDO-->Axis A/B: Velocity mode (vl)	RECORD
0x1A03	0x1A13	Axis A/B-->TPDO: Velocity mode (vl)	RECORD

Date received from the controller (RPDO)

Object		Name	Data type
Axis A	Axis B		
0x6040	0x6840	Controlword	UNSIGNED_16
0x2830	0x3030	Lenze control word	UNSIGNED_16
0x6060	0x6860	Operating mode: Selection	INTEGER_8
0x6042	0x6842	vl target velocity	INTEGER_16

Data transmitted to the controller (TPDO)

Object		Name	Data type
Axis A	Axis B		
0x6041	0x6841	Statusword	UNSIGNED_16
0x2831	0x3031	Lenze status word	UNSIGNED_16
0x6061	0x6861	Modes of operation display	INTEGER_8
0x603F	0x683F	Error code	UNSIGNED_16
0x6044	0x6844	vl velocity actual value	INTEGER_16

7.7.2 Object description

The following two tables provide an overview of the most important objects for this operating mode (without motor parameters, motor control parameters, and feedback parameters).

All objects correspond to the CiA402 specification, but some of them have only a restricted value range.

Objects described in other chapters:

Object		Name	Data type
Axis A	Axis B		
▸ Lenze control and status word			
0x2830	0x3030	Lenze control word	UNSIGNED_16
0x2831	0x3031	Lenze status word	UNSIGNED_16
▸ Device control			
0x6040	0x6840	Controlword	UNSIGNED_16
0x6060	0x6860	Operating mode: Selection	INTEGER_8
0x6041	0x6841	Statusword	UNSIGNED_16
0x6061	0x6861	Modes of operation display	INTEGER_8
Greyed out = read access only			

Objects described in this chapter:

Object		Name	Data type
Axis A	Axis B		
0x6042	0x6842	vl target velocity	INTEGER_16
0x6043	0x6843	vl velocity demand	INTEGER_16
0x6044	0x6844	vl velocity actual value	INTEGER_16
0x6046	0x6846	vl velocity min max amount	RECORD
0x6048	0x6848	vl velocity acceleration	RECORD
0x6049	0x6849	vl velocity deceleration	RECORD

0x6042 | 0x6842 - vl target velocity

Setting range (min. value | unit | max. value)			Lenze setting	
-32768	r/min	32767	0 rpm	
☑ Write access ☐ CINH ☑ OSC ☐ P ☑ RX ☐ TX				INTEGER_16

0x6043 | 0x6843 - vl velocity demand

Display area (min. value | unit | max. value)			Initialisation	
-32768	r/min	32767		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX				INTEGER_16

0x6044 | 0x6844 - vl velocity actual value

Display area (min. value | unit | max. value)			Initialisation	
-32768	r/min	32767		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX				INTEGER_16

0x6046 | 0x6846 - vl velocity min max amount

Sub.	Name	Lenze setting	Data type
1	vl velocity min amount	0 rpm	UNSIGNED_32
2	vl velocity max amount	2147483647 r/min	UNSIGNED_32

Subindex 1: **vl velocity min amount**				
Setting range (min. value | unit | max. value)			Lenze setting	
0	r/min	0	0 rpm	
☑ Write access ☐ CINH ☑ OSC ☐ P ☑ RX ☐ TX				UNSIGNED_32

Subindex 2: **vl velocity max amount**				
Setting range (min. value | unit | max. value)			Lenze setting	
2147483647	r/min	2147483647	2147483647 r/min	
☑ Write access ☐ CINH ☑ OSC ☐ P ☑ RX ☐ TX				UNSIGNED_32

0x6048 | 0x6848 - vl velocity acceleration

Sub.	Name	Lenze setting	Data type
1	Delta speed	0 rpm	UNSIGNED_32
2	Delta time	10 s	UNSIGNED_16

Subindex 1: **Delta speed**				
Setting range (min. value | unit | max. value)			Lenze setting	
0	r/min	2147483647	0 rpm	
☑ Write access ☐ CINH ☑ OSC ☑ P ☑ RX ☐ TX				UNSIGNED_32

Subindex 2: **Delta time**				
Setting range (min. value | unit | max. value)			Lenze setting	
0	s	65535	10 s	
☑ Write access ☐ CINH ☑ OSC ☑ P ☑ RX ☐ TX				UNSIGNED_16

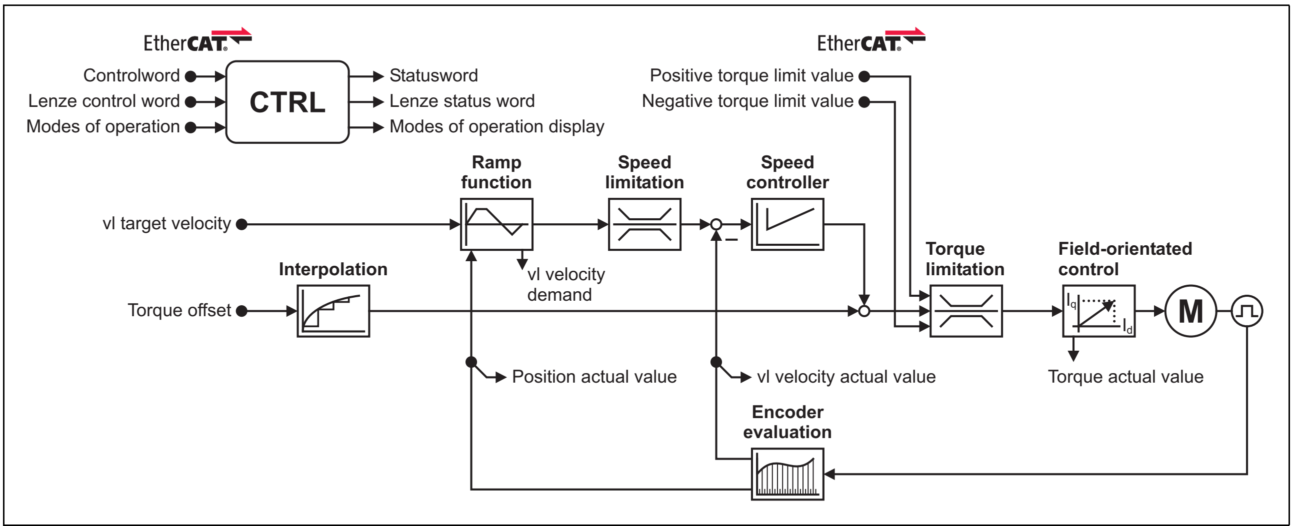
0x6049 | 0x6849 - vl velocity deceleration

Sub.	Name	Lenze setting	Data type
▸ 1	Delta speed	0 rpm	UNSIGNED_32
▸ 2	Delta time	10 s	UNSIGNED_16

Subindex 1: **Delta speed**				
Setting range (min. value | unit | max. value)			Lenze setting	
0	r/min	2147483647	0 rpm	
☑ Write access ☐ CINH ☑ OSC ☑ P ☑ RX ☐ TX				UNSIGNED_32

Subindex 2: **Delta time**				
Setting range (min. value | unit | max. value)			Lenze setting	
0	s	65535	10 s	
☑ Write access ☐ CINH ☑ OSC ☑ P ☑ RX ☐ TX				UNSIGNED_16

7.7.3 Signal flow (servo control)



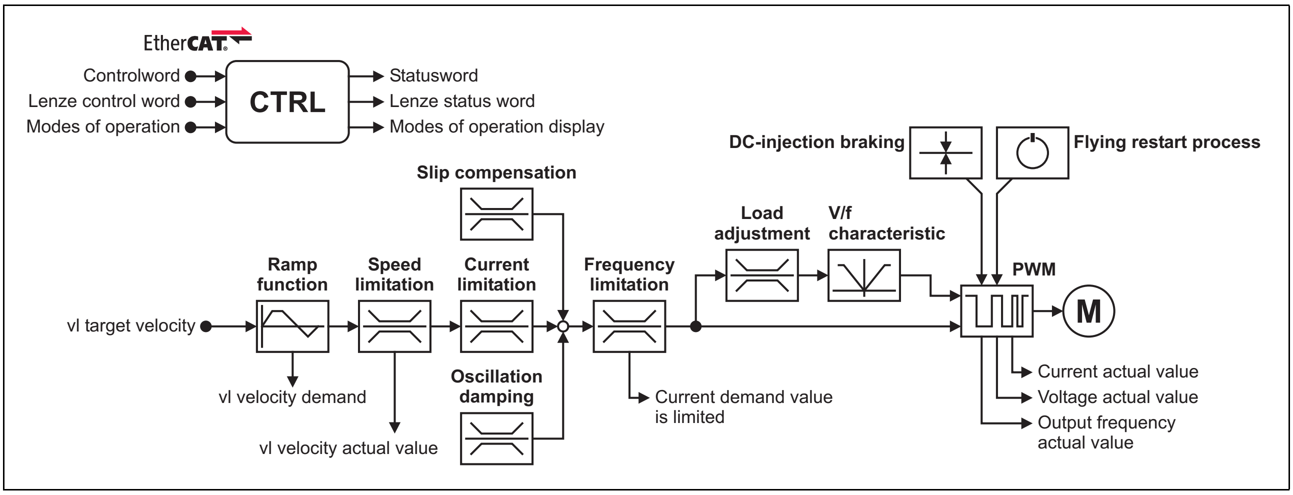
[7-2] Signal flow of the servo control in "Velocity mode" (simplified representation)

Short overview of the most important parameters:

Function	Object		Name
	Axis A	Axis B	
Input data ●—	0x6040	0x6840	Controlword
	0x2830	0x3030	Lenze control word
	0x6060	0x6860	Operating mode: Selection
	0x6042	0x6842	vl target velocity
	0x60B2	0x68B2	Torque offset
	0x60E0	0x68E0	Positive torque limit value
	0x60E1	0x68E1	Negative torque limit value

Function	Object		Name
	Axis A	Axis B	
Output data	0x6041	0x6841	Statusword
	0x2831	0x3031	Lenze status word
	0x6061	0x6861	Modes of operation display
	0x6043	0x6843	vl velocity demand
	0x6064	0x6864	Position actual value
	0x606C	0x686C	Velocity actual value
	0x6077	0x6877	Torque actual value
Interpolation	0x60C2	0x68C2	Interpolation time period
Ramp function	0x6048:1	0x6848:1	Delta speed
	0x6048:2	0x6848:2	Delta time
	0x6049:1	0x6849:1	Delta speed
	0x6049:2	0x6849:2	Delta time
Speed limitation	0x6080	0x6880	Max motor speed
	0x2903	0x3103	Speed: Speed setpoint - filter time
Speed controller	0x2900:1	0x3100:1	Speed controller: Gain
	0x2900:2	0x3100:2	Speed controller: Reset time
	0x2900:3	0x3100:3	Speed controller: Rate time
	0x2901	0x3101	Speed controller: Gain - adaption
Torque limitation	0x60E0	0x68E0	Positive torque limit value
	0x60E1	0x68E1	Negative torque limit value
	0x6076	0x6876	Motor rated torque
	0x6072	0x6872	Max torque
	0x2944:1	0x3144:1	Notch filter 1: Frequency
	0x2944:2	0x3144:2	Notch filter 1: Bandwidth
	0x2944:3	0x3144:3	Notch filter 1: Damping
	0x2944:4	0x3144:4	Notch filter 2: Frequency
	0x2944:5	0x3144:5	Notch filter 2: Bandwidth
	0x2944:6	0x3144:6	Notch filter 2: Damping
Field-oriented control	0x6073	0x6873	Max current
	0x6075	0x6875	Motor rated current
	0x2941	0x3141	Current controller: Feedforward control
	0x2942:1	0x3142:1	Current controller: Gain
	0x2942:2	0x3142:2	Current controller: Reset time
	0x29E2	0x31E2	DC link circuit voltage: Filter time
	0x29E3	0x31E3	Motor voltage act. value: Filter time
	0x29E0:1	0x31E0:1	Field weakening controller: Gain
	0x29E0:2	0x31E0:2	Field weakening controller: Reset time
	0x29E1	0x31E1	Field set value limitation
	0x29C0:1	0x31C0:1	Field controller: Gain
	0x29C0:2	0x31C0:2	Field controller: Reset time
	0x2939	0x3139	Switching frequency

7.7.4 Signal flow (V/f characteristic control)



[7-3] Signal flow of the V/f characteristic control in "Velocity mode" (simplified representation)

Short overview of the most important parameters:

Function	Object		Name
	Axis A	Axis B	
Input data	0x6040	0x6840	Controlword
	0x2830	0x3030	Lenze control word
	0x6060	0x6860	Operating mode: Selection
	0x6042	0x6842	vl target velocity
Output data	0x6041	0x6841	Statusword
	0x2831	0x3031	Lenze status word
	0x6061	0x6861	Modes of operation display
	0x6043	0x6843	vl velocity demand
	0x606C	0x686C	Velocity actual value
	0x6078	0x6878	Current actual value
	0x2D82	0x3582	Motor: Actual voltage - Veff, phase-phase
	0x2DDD	0x35DD	Device: Actual output frequency
Ramp function	0x6048:1	0x6848:1	Delta speed
	0x6048:2	0x6848:2	Delta time
	0x6049:1	0x6849:1	Delta speed
	0x6049:2	0x6849:2	Delta time
Speed limitation	0x6080	0x6880	Max motor speed
	0x2903	0x3103	Speed: Speed setpoint - filter time
Slip compensation	0x2B09:1	0x3309:1	VFC: Slip compensation - influence
	0x2B09:2	0x3309:2	VFC: Slip compensation - filter time
Oscillation damping	0x2B0A:1	0x330A:1	VFC: Oscillation damping - gain
	0x2B0A:2	0x330A:2	VFC: Oscillation damping - filter time
	0x2B0A:3	0x330A:3	VFC: Oscillation damping - limitation
	0x2B0A:4	0x330A:4	VFC: Oscillation damping - ramp-end frequency

Function	Object		Name
	Axis A	Axis B	
Load adjustment	0x2B07:1	0x2B07:1	VFC: Load adjustment - direction of rotation
	0x2B07:2	0x2B07:2	VFC: Load adjustment - value
V/f characteristic	0x2B01:1	0x3301:1	VFC: V/f characteristic - voltage at reference point
	0x2B01:2	0x3301:2	VFC: V/f characteristic - frequency at reference point
	0x2B06	0x3306	VFC: Voltage boost
	0x2B04	0x3304	VFC: Voltage vector control - current setpoint
	0x2B00	0x3300	VFC: V/f characteristic - shape
	0x2B02:x	0x3302:x	VFC: User defined V/f characteristic • Frequency grid points (x1 ... x11)
	0x2B03:x	0x3303:x	VFC: User defined V/f characteristic • Voltage grid points (y1 ... y11)
DC-injection braking	0x2B80	0x3380	DC-injection braking: Current
Flying restart function	0x2BA0	0x33A0	Flying restart: Activate
	0x2BA1	0x33A1	Flying restart: Current
	0x2BA2	0x33A2	Flying restart: Start frequency
	0x2BA3	0x33A3	Flying restart: Integration time
	0x2BA4	0x33A4	Flying restart: Min. deviation
	0x2BA5	0x33A5	Flying restart: Delay time
	0x2BA6	0x33A6	Flying restart: Result

Tip!

A detailed representation of the signal flow with all relevant parameters can be found in the »PLC Designer« on the **Signal flow** tab for the i700 servo inverter.

7.8 Cyclic sync position mode (csp)

This operating mode provides a quick position follower with speed/torque/feed force feedforward control. The motion profile to be processed is defined by the controller.

Note!

This operating mode can only be actuated reasonably by the use of a servo control.

▶ Motor control & motor settings

Subfunctions of the operating mode

- Calculation of the position setpoint
- Calculation of the speed feedforward control value
- Calculation of the speed feedforward control value
- Interpolation between communication cycle and control cycle
- Position control
- Speed control
- Torque control
- Update of the actual values for position, speed, and torque

Selection of the operating mode

"Cyclic sync position mode" is selected with the setting "8" in 0x6060 (or 0x6860 for axis B).

7.8.1 Default mapping

The default mapping for the cyclic sync position mode is defined in the following objects:

Object		Name	Data type
Axis A	Axis B		
0x1600	0x1610	RPDO-->Axis A/B: Cyclic syncs position (csp)	RECORD
0x1606	0x1616	RPDO-->Axis A/B: Torque limits	RECORD
0x1A00	0x1A10	Axis A/B-->TPDO: Cyclic sync position (csp)	RECORD

Date received from the controller (RPDO)

Object		Name	Data type
Axis A	Axis B		
0x6040	0x6840	Controlword	UNSIGNED_16
0x2830	0x3030	Lenze control word	UNSIGNED_16
0x6060	0x6860	Operating mode: Selection	INTEGER_8
0x60B2	0x68B2	Torque offset	INTEGER_16
0x607A	0x687A	Position demand value	INTEGER_32
0x60B1	0x68B1	Velocity offset	INTEGER_32
0x2902	0x3102	Speed controller: Load value	INTEGER_16
0x60E0	0x68E0	Positive torque limit value	UNSIGNED_16
0x60E1	0x68E1	Negative torque limit value	UNSIGNED_16

Data transmitted to the controller (TPDO)

Object		Name	Data type
Axis A	Axis B		
0x6041	0x6841	Statusword	UNSIGNED_16
0x2831	0x3031	Lenze status word	UNSIGNED_16
0x6061	0x6861	Modes of operation display	INTEGER_8
0x603F	0x683F	Error code	UNSIGNED_16
0x606C	0x686C	Velocity actual value	UNSIGNED_16
0x6077	0x6877	Torque actual value	INTEGER_16
0x6064	0x6864	Position actual value	INTEGER_32
0x60F4	0x68F4	Following error actual value	INTEGER_32

7.8.2 Object description

The following two tables provide an overview of the most important objects for this operating mode (without motor parameters, motor control parameters, and feedback parameters).

All objects correspond to the CiA402 specification, but some of them have only a restricted value range.

Objects described in other chapters:

Object Axis A	 Axis B	Name	Data type
Lenze control and status word			
0x2830	0x3030	Lenze control word	UNSIGNED_16
0x2831	0x3031	Lenze status word	UNSIGNED_16
Device control			
0x6040	0x6840	Controlword	UNSIGNED_16
0x6060	0x6860	Modes of operation	INTEGER_8
0x6041	0x6841	Statusword	UNSIGNED_16
0x6061	0x6861	Modes of operation display	INTEGER_8
Parameters for actuation of the position control			
0x6065	0x6865	Following error window	UNSIGNED_32
0x6066	0x6866	Following error time out	UNSIGNED_16
0x6067	0x6867	Position window	UNSIGNED_32
0x6068	0x6868	Position window time	UNSIGNED_16
0x6062	0x6862	Position demand value	INTEGER_32
0x6063	0x6863	Position actual internal value	INTEGER_32
0x6064	0x6864	Position actual value	INTEGER_32
0x60F4	0x68F4	Following error actual value	INTEGER_32
0x60FC	0x68FC	Position demand internal value	INTEGER_32
Cyclic sync velocity mode (csv)			
0x60B1	0x68B1	Velocity offset	INTEGER_32
0x60FF	0x68FF	Target velocity	INTEGER_32
0x606C	0x686C	Velocity actual value	UNSIGNED_16
Cyclic sync torque mode (cst)			
0x60B2	0x68B2	Torque offset	INTEGER_16
0x6071	0x6871	Target torque	INTEGER_16
0x6077	0x6877	Torque actual value	INTEGER_16
Greyed out = read access only			

Objects described in this chapter:

Object Axis A	 Axis B	Name	Data type
0x607A	0x687A	Position demand value	INTEGER_32
0x60C0	0x68C0	Interpolation sub mode select	INTEGER_16
0x60C2	0x68C2	Interpolation time period	RECORD
0x60E0	0x68E0	Positive torque limit value	UNSIGNED_16
0x60E1	0x68E1	Negative torque limit value	UNSIGNED_16

0x607A | 0x687A - Target position

Setting range (min. value | unit | max. value)			Lenze setting	
-2147483648	[Pos unit]	2147483647	0 [Pos unit]	
☑ Write access ☐ CINH ☑ OSC ☐ P ☑ RX ☐ TX				INTEGER_32

0x60C0 | 0x68C0 - Interpolation sub mode select

From version 01.03
Interpolation sub mode select
0 = linear position interpolation
-1 = square position interpolation
All other values have no function and are rejected.

Setting range (min. value | unit | max. value)			Lenze setting	
-32768		32767	0	
☑ Write access ☐ CINH ☐ OSC ☑ P ☑ RX ☐ TX				INTEGER_16

0x60C2 | 0x68C2 - Interpolation time period

If a Lenze Controller is used, the interpolation time interval is automatically set to the time given in the »PLC Designer« as EtherCAT master cycle time.
- Only if a controller of a third-party manufacturer is used: Set the interpolation time interval used for the process data communication in this object.
- Preset interpolation time interval = 1 * 10^{-3} s = 1 ms

Sub.	Name	Lenze setting	Data type
1	Interpolation time period	1 10^(Interpolation time index) s	UNSIGNED_8
2	Interpolation time index	-3	INTEGER_8

Subindex 1: **Interpolation time period**				
Basic multiplier for interpolation time interval				
Setting range (min. value | unit | max. value)			Lenze setting	
0	10^(Interpolation time index) s	255	1 10^(Interpolation time index) s	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_8

Subindex 2: **Interpolation time index**				
Exponent for interpolation time period				
Setting range (min. value | unit | max. value)			Lenze setting	
-6		0	-3	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				INTEGER_8

$$\text{Interpolation time period}[s] = \text{Time period value} \cdot 10^{\text{Index}}[s]$$

[7-4] Definition of the interpolation time period

Tip!

For an interpolation cycle of 2 ms, for instance, the following values are to be set:
- Subindex 1 (Time period value) = "2"
- Subindex 2 (Time index) = "-3"

0x60E0 | 0x68E0 - Positive torque limit value

100 % ≡ rated motor torque (0x6076 or 0x6876 for axis B)

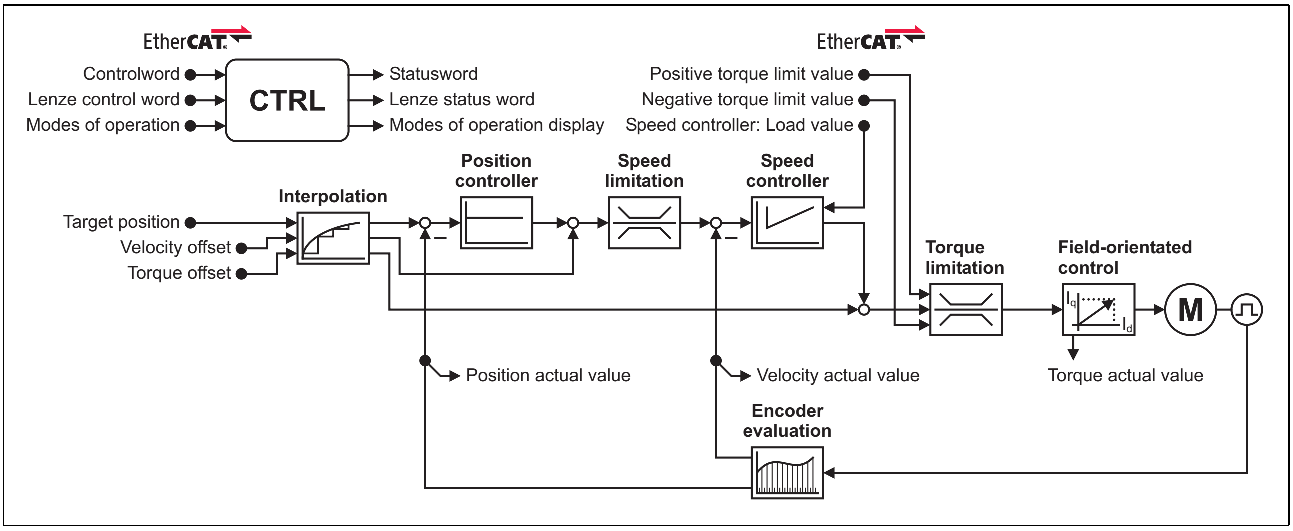
Setting range (min. value | unit | max. value)			Lenze setting	
0.0	%	3276.7	100.0 %	
☑ Write access ☐ CINH ☑ OSC ☑ P ☑ RX ☐ TX			Scaling: 1/10	UNSIGNED_16

0x60E1 | 0x68E1 - Negative torque limit value

100 % ≡ rated motor torque (0x6076 or 0x6876 for axis B)

Setting range (min. value | unit | max. value)			Lenze setting	
0.0	%	3276.7	100.0 %	
☑ Write access ☐ CINH ☑ OSC ☑ P ☑ RX ☐ TX			Scaling: 1/10	UNSIGNED_16

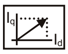
7.8.3 Signal flow



[7-5] Signal flow of the servo control in cyclic sync position mode (simplified representation)

Short overview of the most important parameters:

Function	Object		Name
	Axis A	Axis B	
Input data	0x6040	0x6840	Controlword
	0x2830	0x3030	Lenze control word
	0x6060	0x6860	Operating mode: Selection
	0x607A	0x687A	Position demand value
	0x60B1	0x68B1	Velocity offset
	0x60B2	0x68B2	Torque offset
	0x60E0	0x68E0	Positive torque limit value
	0x60E1	0x68E1	Negative torque limit value
	0x2902	0x3102	Speed controller: Load value
Output data	0x6041	0x6841	Statusword
	0x2831	0x3031	Lenze status word
	0x6061	0x6861	Modes of operation display
	0x6064	0x6864	Position actual value
	0x606C	0x686C	Velocity actual value
	0x6077	0x6877	Torque actual value
Interpolation	0x60C0	0x68C0	Interpolation sub mode select
	0x60C2	0x68C2	Interpolation time period
Position controller	0x2980	0x3180	Position controller: Gain
	0x2981	0x3181	Position controller: Gain - adaption
	0x2982	0x3182	Position controller: Output signal limitation
Speed limitation	0x6080	0x6880	Max motor speed
	0x2903	0x3103	Speed: Speed setpoint - filter time
Speed controller	0x2900:1	0x3100:1	Speed controller: Gain
	0x2900:2	0x3100:2	Speed controller: Reset time
	0x2900:3	0x3100:3	Speed controller: Rate time
	0x2901	0x3101	Speed controller: Gain - adaption
Torque limitation	0x60E0	0x68E0	Positive torque limit value
	0x60E1	0x68E1	Negative torque limit value
	0x6076	0x6876	Motor rated torque
	0x6072	0x6872	Max torque
	0x2944:1	0x3144:1	Notch filter 1: Frequency
	0x2944:2	0x3144:2	Notch filter 1: Bandwidth
	0x2944:3	0x3144:3	Notch filter 1: Damping
	0x2944:4	0x3144:4	Notch filter 2: Frequency
	0x2944:5	0x3144:5	Notch filter 2: Bandwidth
	0x2944:6	0x3144:6	Notch filter 2: Damping

Function	Object		Name
	Axis A	Axis B	
Field-oriented control 	0x6073	0x6873	Max current
	0x6075	0x6875	Motor rated current
	0x2941	0x3141	Current controller: Feedforward control
	0x2942:1	0x3142:1	Current controller: Gain
	0x2942:2	0x3142:2	Current controller: Reset time
	0x29E2	0x31E2	DC link circuit voltage: Filter time
	0x29E3	0x31E3	Motor voltage act. value: Filter time
	0x29E0:1	0x31E0:1	Field weakening controller: Gain
	0x29E0:2	0x31E0:2	Field weakening controller: Reset time
	0x29E1	0x31E1	Field set value limitation
	0x29C0:1	0x31C0:1	Field controller: Gain
	0x29C0:2	0x31C0:2	Field controller: Reset time
	0x2939	0x3139	Switching frequency

Tip!

A detailed representation of the signal flow with all relevant parameters can be found in the »PLC Designer« on the **Signal flow** tab for the i700 servo inverter.

7.8.4 Control commands & status information

The following control commands can be executed in cyclic sync position mode via the Controlword (0x6040 or 0x6840 for axis B):

Control word	Status	Function
Bit 4	0	*Reserved* (bit must be set to "0")
Bit 5	0	*Reserved* (bit must be set to "0")
Bit 6	0	*Reserved* (bit must be set to "0")
Bit 8	0 ↗ 1	Stop

The following status information is output in cyclic sync position mode via the Statusword (0x6041 or 0x6841 for axis B):

Status word	Status	Meaning
Bit 12	0	Cyclic sync position mode inactive
	1	Cyclic sync position mode active

7.9 Cyclic sync velocity mode (csv)

This operating mode provides a quick speed follower with torque/feed force feedforward control. The motion profile to be processed is defined by the controller.

Subfunctions of the operating mode

- Calculation of the speed setpoint
- Calculation of the speed feedforward control value
- Interpolation between communication cycle and control cycle
- Speed control
- Torque control
- Limitation of the motor speed
- Update of the actual values for position, speed, and torque

Selection of the operating mode

Cyclic sync velocity mode is selected with the setting "9" in 0x6060 (or 0x6860 for axis B).

7.9.1 Default mapping

The default mapping for the cyclic sync velocity mode is defined in the following objects:

Object		Name	Data type
Axis A	Axis B		
0x1602	0x1612	RPDO-->Axis A/B: Cyclic sync velocity mode (csv)	RECORD
0x1606	0x1616	RPDO-->Axis A/B: Torque limits	RECORD
0x1A02	0x1A12	Axis A/B-->TPDO: Cyclic sync velocity mode (csv)	RECORD

Date received from the controller (RPDO)

Object		Name	Data type
Axis A	Axis B		
0x6040	0x6840	Controlword	UNSIGNED_16
0x2830	0x3030	Lenze control word	UNSIGNED_16
0x6060	0x6860	Operating mode: Selection	INTEGER_8
0x60B2	0x68B2	Torque offset	INTEGER_16
0x60FF	0x68FF	Target velocity	INTEGER_32
0x60E0	0x68E0	Positive torque limit value	UNSIGNED_16
0x60E1	0x68E1	Negative torque limit value	UNSIGNED_16

Data transmitted to the controller (TPDO)

Object		Name	Data type
Axis A	Axis B		
0x6041	0x6841	Statusword	UNSIGNED_16
0x2831	0x3031	Lenze status word	UNSIGNED_16
0x6061	0x6861	Modes of operation display	INTEGER_8
0x603F	0x683F	Error code	UNSIGNED_16
0x606C	0x686C	Velocity actual value	UNSIGNED_16
0x6077	0x6877	Torque actual value	INTEGER_16
0x6064	0x6864	Position actual value	INTEGER_32

7.9.2 Object description

The following two tables provide an overview of the most important objects for this operating mode (without motor parameters, motor control parameters, and feedback parameters).

All objects correspond to the CiA402 specification, but some of them have only a restricted value range.

Objects described in other chapters:

Object Axis A	 Axis B	Name	Data type
Lenze control and status word			
0x2830	0x3030	Lenze control word	UNSIGNED_16
0x2831	0x3031	Lenze status word	UNSIGNED_16
Device control			
0x6040	0x6840	Controlword	UNSIGNED_16
0x6060	0x6860	Operating mode: Selection	INTEGER_8
0x6041	0x6841	Statusword	UNSIGNED_16
0x6061	0x6861	Modes of operation display	INTEGER_8
Parameters for actuation of the position control			
0x6064	0x6864	Position actual value	INTEGER_32
Cyclic sync position mode (csp)			
0x60C2	0x68C2	Interpolation time period	RECORD
0x60E0	0x68E0	Positive torque limit value	UNSIGNED_16
0x60E1	0x68E1	Negative torque limit value	UNSIGNED_16
Cyclic sync torque mode (cst)			
0x60B2	0x68B2	Torque offset	INTEGER_16
0x6071	0x6871	Target torque	INTEGER_16
0x6077	0x6877	Torque actual value	INTEGER_16
Greyed out = read access only			

Objects described in this chapter:

Object Axis A	 Axis B	Name	Data type
0x606C	0x686C	Velocity actual value	INTEGER_32
0x60B1	0x68B1	Velocity offset	INTEGER_32
0x60FF	0x68FF	Target velocity	INTEGER_32
Greyed out = read access only			

0x606C | 0x686C - Velocity actual value

Current speed

Display area (min. value | unit | max. value)			Initialisation	
-480000	[n unit]	480000		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: $480000/2^{31}$	INTEGER_32

0x60B1 | 0x68B1 - Velocity offset

Additive value for target velocity or velocity feedforward control

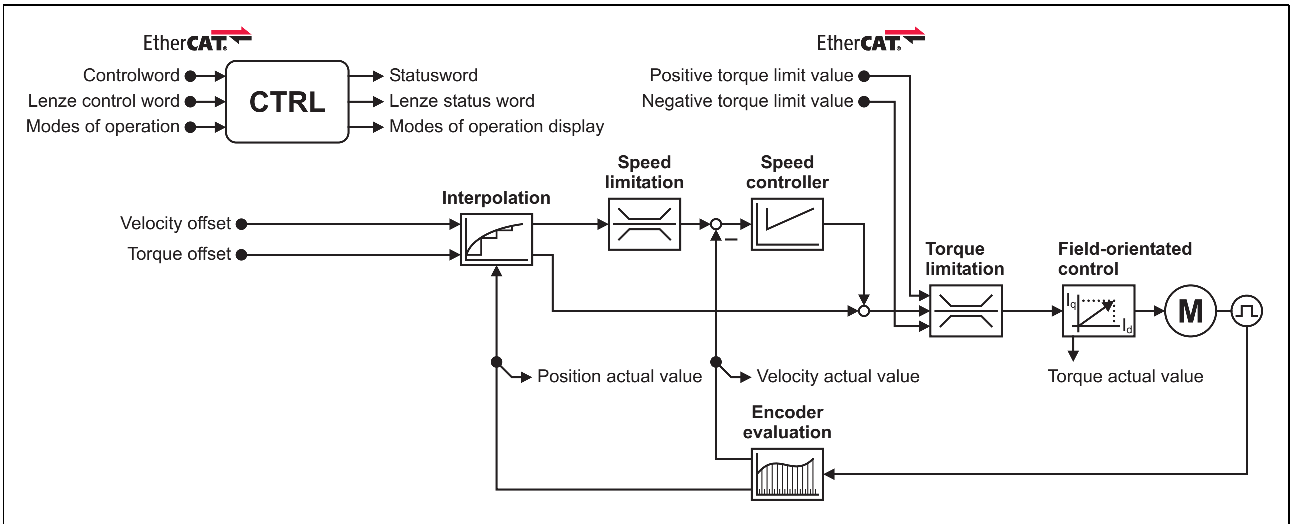
Setting range (min. value | unit | max. value)			Lenze setting	
-480000	[n unit]	480000	0 [n-unit]	
☑ Write access ☐ CINH ☑ OSC ☑ P ☑ RX ☐ TX			Scaling: $480000/2^{31}$	INTEGER_32

0x60FF | 0x68FF - Target velocity

Target velocity or velocity feedforward control

Setting range (min. value | unit | max. value)			Lenze setting	
-480000	[n unit]	480000	0 [n-unit]	
☑ Write access ☐ CINH ☑ OSC ☐ P ☑ RX ☐ TX			Scaling: $480000/2^{31}$	INTEGER_32

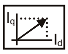
7.9.3 Signal flow (servo control)



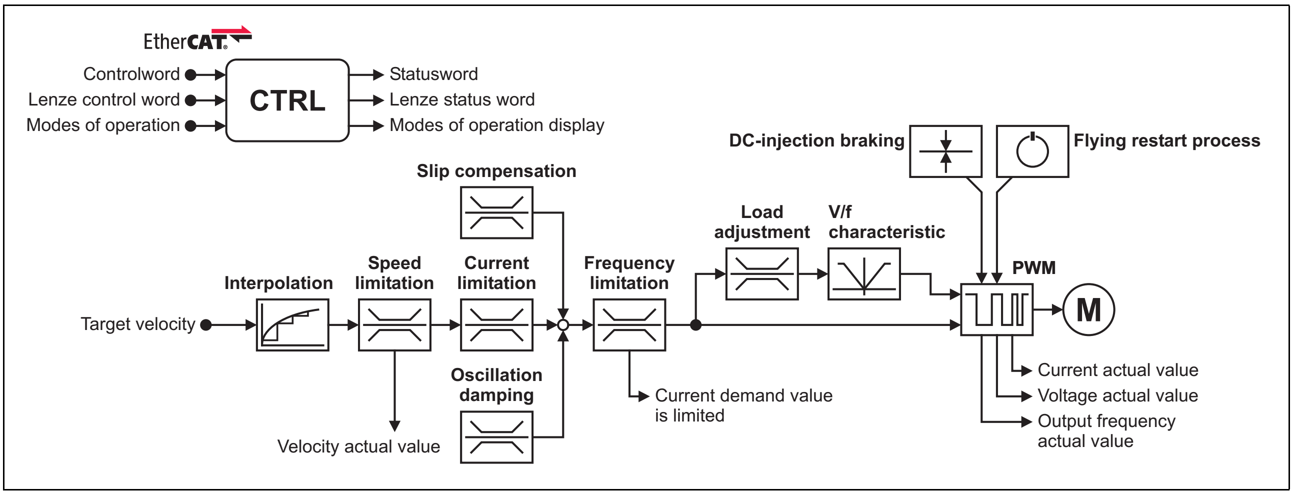
[7-6] Signal flow of the servo control in cyclic sync velocity mode (simplified representation)

Short overview of the most important parameters:

Function	Object		Name
	Axis A	Axis B	
Input data	0x6040	0x6840	Controlword
	0x2830	0x3030	Lenze control word
	0x6060	0x6860	Operating mode: Selection
	0x60B1	0x68B1	Velocity offset
	0x60B2	0x68B2	Torque offset
	0x60E0	0x68E0	Positive torque limit value
	0x60E1	0x68E1	Negative torque limit value
Output data	0x6041	0x6841	Statusword
	0x2831	0x3031	Lenze status word
	0x6061	0x6861	Modes of operation display
	0x6064	0x6864	Position actual value
	0x606C	0x686C	Velocity actual value
	0x6077	0x6877	Torque actual value
Interpolation	0x60C2	0x68C2	Interpolation time period
Speed limitation	0x6080	0x6880	Max motor speed
	0x2903	0x3103	Speed: Speed setpoint - filter time
Speed controller	0x2900:1	0x3100:1	Speed controller: Gain
	0x2900:2	0x3100:2	Speed controller: Reset time
	0x2900:3	0x3100:3	Speed controller: Rate time
	0x2901	0x3101	Speed controller: Gain - adaption
Torque limitation	0x60E0	0x68E0	Positive torque limit value
	0x60E1	0x68E1	Negative torque limit value
	0x6076	0x6876	Motor rated torque
	0x6072	0x6872	Max torque
	0x2944:1	0x3144:1	Notch filter 1: Frequency
	0x2944:2	0x3144:2	Notch filter 1: Bandwidth
	0x2944:3	0x3144:3	Notch filter 1: Damping
	0x2944:4	0x3144:4	Notch filter 2: Frequency
	0x2944:5	0x3144:5	Notch filter 2: Bandwidth
	0x2944:6	0x3144:6	Notch filter 2: Damping

Function	Object		Name
	Axis A	Axis B	
Field-oriented control 	0x6073	0x6873	Max current
	0x6075	0x6875	Motor rated current
	0x2941	0x3141	Current controller: Feedforward control
	0x2942:1	0x3142:1	Current controller: Gain
	0x2942:2	0x3142:2	Current controller: Reset time
	0x29E2	0x31E2	DC link circuit voltage: Filter time
	0x29E3	0x31E3	Motor voltage act. value: Filter time
	0x29E0:1	0x31E0:1	Field weakening controller: Gain
	0x29E0:2	0x31E0:2	Field weakening controller: Reset time
	0x29E1	0x31E1	Field set value limitation
	0x29C0:1	0x31C0:1	Field controller: Gain
	0x29C0:2	0x31C0:2	Field controller: Reset time
	0x2939	0x3139	Switching frequency

7.9.4 Signal flow (V/f characteristic control)



[7-7] Signal flow of the V/f characteristic control in cyclic sync velocity mode (simplified representation)

Short overview of the most important parameters:

Function	Object		Name
	Axis A	Axis B	
Input data	0x6040	0x6840	Controlword
	0x2830	0x3030	Lenze control word
	0x6060	0x6860	Operating mode: Selection
	0x60FF	0x68FF	Target velocity

Function	Object		Name
	Axis A	Axis B	
Output data	0x6041	0x6841	Statusword
	0x2831	0x3031	Lenze status word
	0x6061	0x6861	Modes of operation display
	0x606C	0x686C	Velocity actual value
	0x6078	0x6878	Current actual value
	0x2D82	0x3582	Motor: Actual voltage - Veff, phase-phase
	0x2DDD	0x35DD	Device: Actual output frequency
Interpolation	0x60C2	0x68C2	Interpolation time period
Speed limitation	0x6080	0x6880	Max motor speed
	0x2903	0x3103	Speed: Speed setpoint - filter time
Slip compensation	0x2B09:1	0x3309:1	VFC: Slip compensation - influence
	0x2B09:2	0x3309:2	VFC: Slip compensation - filter time
Oscillation damping	0x2B0A:1	0x330A:1	VFC: Oscillation damping - gain
	0x2B0A:2	0x330A:2	VFC: Oscillation damping - filter time
	0x2B0A:3	0x330A:3	VFC: Oscillation damping - limitation
	0x2B0A:4	0x330A:4	VFC: Oscillation damping - ramp-end frequency
Load adjustment	0x2B07:1	0x2B07:1	VFC: Load adjustment - direction of rotation
	0x2B07:2	0x2B07:2	VFC: Load adjustment - value
V/f characteristic	0x2B01:1	0x3301:1	VFC: V/f characteristic - voltage at reference point
	0x2B01:2	0x3301:2	VFC: V/f characteristic - frequency at reference point
	0x2B06	0x3306	VFC: Voltage boost
	0x2B04	0x3304	VFC: Voltage vector control - current setpoint
	0x2B00	0x3300	VFC: V/f characteristic - shape
	0x2B02:x	0x3302:x	VFC: User defined V/f characteristic • Frequency grid points (x1 ... x11)
	0x2B03:x	0x3303:x	VFC: User defined V/f characteristic • Voltage grid points (y1 ... y11)
DC-injection braking	0x2B80	0x3380	DC-injection braking: Current
Flying restart function	0x2BA0	0x33A0	Flying restart: Activate
	0x2BA1	0x33A1	Flying restart: Current
	0x2BA2	0x33A2	Flying restart: Start frequency
	0x2BA3	0x33A3	Flying restart: Integration time
	0x2BA4	0x33A4	Flying restart: Min. deviation
	0x2BA5	0x33A5	Flying restart: Delay time
	0x2BA6	0x33A6	Flying restart: Result

Tip!

A detailed representation of the signal flow with all relevant parameters can be found in the »PLC Designer« on the **Signal flow** tab for the i700 servo inverter.

7.9.5 Control commands & status information

The following control commands can be executed in "Cyclic synchronous velocity mode" via the Controlword (0x6040 or 0x6840 for axis B):

Control word	Status	Function
Bit 4	0	*Reserved* (bit must be set to "0")
Bit 5	0	*Reserved* (bit must be set to "0")
Bit 6	0	*Reserved* (bit must be set to "0")
Bit 8	0 ↗ 1	Stop

The following status information is output in "Cyclic synchronous velocity mode" via the Statusword (0x6041 or 0x6841 for axis B):

Status word	Status	Meaning
Bit 12	0	Cyclic sync velocity mode inactive
	1	Cyclic sync velocity mode active

7.10 Cyclic sync torque mode (cst)

This operating mode provides a quick torque follower with speed limitation. The torque profile to be processed is defined by the controller.

Note!

- This operating mode can only be actuated reasonably if the servo control is set as motor control. ▶ Motor control & motor settings
- Since this is a cyclic mode, setpoints in the bus cycle are expected.
- If communication is interrupted and thus telegrams fail to appear, the internal "quick stop" function takes over the braking of the motor. Afterwards, the controller is inhibited.

Subfunctions of the operating mode

- Torque control with speed limitation
- Calculation of the torque setpoint
- Interpolation of the setpoints from the communication cycle to the control cycle
- Limitation of the motor speed
- Update of the actual values for position, speed, and torque

Selection of the operating mode

"Cyclic synchronous torque mode" is selected with the setting "10" in 0x6060 (or 0x6860 for axis B).

7.10.1 Default mapping

The default mapping for "Cyclic synchronous torque mode" is defined in the following objects:

Object		Name	Data type
Axis A	Axis B		
0x1601	0x1611	RPDO-->Axis A/B: Cyclic sync torque mode (cst)	RECORD
0x1A01	0x1A11	Axis A/B-->TPDO: Cyclic sync torque mode (cst)	RECORD

Date received from the controller (RPDO)

Object		Name	Data type
Axis A	Axis B		
0x6040	0x6840	Controlword	UNSIGNED_16
0x2830	0x3030	Lenze control word	UNSIGNED_16
0x6060	0x6860	Operating mode: Selection	INTEGER_8
0x60B2	0x68B2	Torque offset	INTEGER_16
0x6071	0x6871	Target torque	INTEGER_16
0x2946:1	0x3146:1	Speed limitation: Upper speed limit	INTEGER_32
0x2946:2	0x3146:2	Speed limitation: Lower speed limit	INTEGER_32

Data transmitted to the controller (TPDO)

Object		Name	Data type
Axis A	Axis B		
0x6041	0x6841	Statusword	UNSIGNED_16
0x2831	0x3031	Lenze status word	UNSIGNED_16
0x6061	0x6861	Modes of operation display	INTEGER_8
0x603F	0x683F	Error code	UNSIGNED_16
0x606C	0x686C	Velocity actual value	UNSIGNED_16
0x6077	0x6877	Torque actual value	INTEGER_16

7.10.2 Object description

The following two tables provide an overview of the most important objects for this operating mode (without motor parameters, motor control parameters, and feedback parameters).

All objects correspond to the CiA402 specification, but some of them have only a restricted value range.

Objects described in other chapters:

Object Axis A	Object Axis B	Name	Data type
Lenze control and status word			
0x2830	0x3030	Lenze control word	UNSIGNED_16
0x2831	0x3031	Lenze status word	UNSIGNED_16
Device control			
0x6040	0x6840	Controlword	UNSIGNED_16
0x6060	0x6860	Operating mode: Selection	INTEGER_8
0x6041	0x6841	Statusword	UNSIGNED_16
0x6061	0x6861	Modes of operation display	INTEGER_8
Parameters for actuation of the position control			
0x6064	0x6864	Position actual value	INTEGER_32
Cyclic sync position mode (csp)			
0x60C2	0x68C2	Interpolation time period	RECORD
Cyclic sync velocity mode (csv)			
0x606C	0x686C	Velocity actual value	INTEGER_32
Set motor parameters manually			
0x6075	0x6875	Motor rated current	UNSIGNED_32
0x6076	0x6876	Motor rated torque	UNSIGNED_32
Greyed out = read access only			

Objects described in this chapter:

Object Axis A	Object Axis B	Name	Data type
0x2946	0x3146	Cyclic sync torque mode: Speed limitation	RECORD
0x6071	0x6871	Target torque	INTEGER_16
0x6072	0x6872	Max torque	UNSIGNED_16
0x6073	0x6873	Max current	UNSIGNED_16
0x6074	0x6874	Target torque	INTEGER_16
0x6077	0x6877	Torque actual value	INTEGER_16
0x6078	0x6878	Current actual value	INTEGER_16
0x6079	0x6879	DC bus: Actual voltage	UNSIGNED_32
0x60B2	0x68B2	Torque offset	INTEGER_16
Greyed out = read access only			

0x2946 | 0x3146 - Cyclic sync torque mode: Speed limitation

The torque control contains a speed limitation as a protective function against very high speeds. High speeds can occur in the case of a mere torque selection if no counter-torque is available (load-free machine).
The speed limitation function takes effect if the actual motor speed reaches the limit values set here: The motor speed is kept at the limit value in each case. If the machine is decelerated by a counter-torque again, the speed limitation hands control back to the external setpoint via the setpoint torque.
Note!
The upper limit value must be set to a greater value than the lower limit value. There is no plausibility check in the i700 servo inverter.

Sub.	Name	Lenze setting	Data type
▸ 1	Speed limitation: Upper speed limit	0 [n-unit]	INTEGER_32
▸ 2	Speed limitation: Lower speed limit	0 [n-unit]	INTEGER_32

Subindex 1: **Speed limitation: Upper speed limit**				
Upper limit for the speed limitation				
Setting range (min. value | unit | max. value)			Lenze setting	
-480000	[n unit]	480000	0 [n-unit]	
☑ Write access ☐ CINH ☑ OSC ☑ P ☑ RX ☐ TX			Scaling: $480000/2^{31}$	INTEGER_32

Subindex 2: **Speed limitation: Lower speed limit**				
Lower limit for the speed limitation				
Setting range (min. value | unit | max. value)			Lenze setting	
-480000	[n unit]	480000	0 [n-unit]	
☑ Write access ☐ CINH ☑ OSC ☑ P ☑ RX ☐ TX			Scaling: $480000/2^{31}$	INTEGER_32

0x6071 | 0x6871 - Torque demand

Target torque or torque feedforward control
- 100 % ≡ rated motor torque (0x6076 or 0x6876 for axis B)

Setting range (min. value | unit | max. value)			Lenze setting	
-3276.8	%	3276.7	0.0 %	
☑ Write access ☐ CINH ☑ OSC ☐ P ☑ RX ☐ TX			Scaling: 1/10	INTEGER_16

0x6072 | 0x6872 - Max torque

Symmetrical selection of the maximum permissible torque
- 100 % ≡ rated motor torque (0x6076 or 0x6876 for axis B)
- This parameter serves to implement a statically and bipolarly acting torque limitation. This can be used, for instance, as overload protection of the mechanical transmission path/elements starting at the motor shaft.
- This limitation acts independently of the unipolarly acting torque limitations which act as process data on the objects 0x60E0 (or 0x60E0 for axis B) and 0x60E1 (or 0x60E1 for axis B). For details see signal flow.

Setting range (min. value | unit | max. value)			Lenze setting	
0.0	%	3276.7	250.0 %	
☑ Write access ☐ CINH ☑ OSC ☑ P ☑ RX ☐ TX			Scaling: 1/10	UNSIGNED_16

0x6073 | 0x6873 - Max current

Maximum current (maximum overload current of the device)

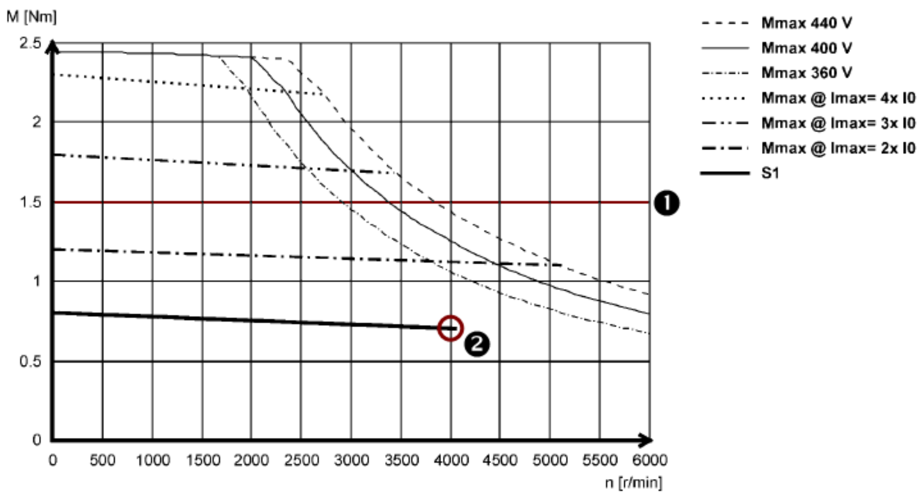
- 100 % ≡ rated motor current (0x6075 or 0x6875 for axis B)
- This parameter serves to limit the inverter output current.
- The value to be set here results from the maximum torque required for the process and the connected motor. The value is calculated according to the following rule of thumb:
  - For synchronous motors (SM):
    Axis A: 0x6073 [%] = $M_{max\_process}$ [Nm] / 0x6076 [Nm] * 100
    Axis B: 0x6873 [%] = $M_{max\_process}$ [Nm] / 0x6876 [Nm] * 100
  - For asynchronous motors (ASM):
    Axis A: 0x6073 [%] = $M_{max\_process}$ [Nm] / 0x6076 [Nm] * 115
    Axis B: 0x6873 [%] = $M_{max\_process}$ [Nm] / 0x6876 [Nm] * 115
  - For both motor types:
    Axis A: 0x6072 [%] = $M_{max\_process}$ [Nm] / 0x6076 [Nm] * 100
    Axis B: 0x6872 [%] = $M_{max\_process}$ [Nm] / 0x6876 [Nm] * 100
- Depending on the motor frame size, the Lenze setting of 0x6073 (or 0x6873 for axis B) and 0x6072 (or 0x6872 for axis B) serves to achieve a rated current of 150 % in case of synchronous motors and a rated current of 130 % in case of asynchronous motors. If the process (application) needs torques above these limits, these two parameters have to be set accordingly based on the motor data.
- If a value is set here which, multiplied by the rated motor current (0x6075 or 0x6875 for axis B) leads to a higher current value than the maximum device current, the value is limited device-internally to a maximum device current / rated motor current of 100 % * and output as warning in the logbook. The value set here, however, is not changed.

Note!

This parameter is not identical to the ultimate motor current I_{ULT} (0x2D46:1 or 0x3546:1 for axis B!

- The ultimate motor current is a limit value for synchronous motors in order to protect their magnets.
- The value to be set here should always be considerably below the ultimate motor current!

Setting range (min. value | unit | max. value)			Lenze setting	
0.0	%	3276.7	150.0 %	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	UNSIGNED_16

Example calculation based on the torque characteristics for MCS06C41 Lenze motor		
The data apply to a mains voltage 3 x 400 V		
		
❶	$M_{max\_process}$	0x6073 = $M_{max\_process}$ / M_{rated_motor} * 100 % = 1.5 Nm / 0.7 Nm * 100 % = 215 %
❷	M_{rated_motor}	

Tip!
More torque characteristics can be found on the Internet:
http://www.lenze.com/dsc

0x6074 | 0x6874 - Torque demand

Interpolated target torque
- 100 % ≡ rated motor torque (0x6076 or 0x6876 for axis B)

Display area (min. value | unit | max. value)			Initialisation	
-3276.8	%	3276.7	0.0 %	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☐ TX			Scaling: 1/10	INTEGER_16

0x6077 | 0x6877 - Torque actual value

Current torque
- 100 % ≡ rated motor torque (0x6076 or 0x6876 for axis B)

Display area (min. value | unit | max. value)			Initialisation	
-3276.8	%	3276.7		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/10	INTEGER_16

0x6078 | 0x6878 - Current actual value

Actual motor current
- 100 % ≡ rated motor current (0x6075 or 0x6875 for axis B)

Display area (min. value | unit | max. value)			Initialisation	
-3276.8	%	3276.7		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/10	INTEGER_16

0x6079 | 0x6879 - DC link circuit voltage

Actual DC-bus voltage

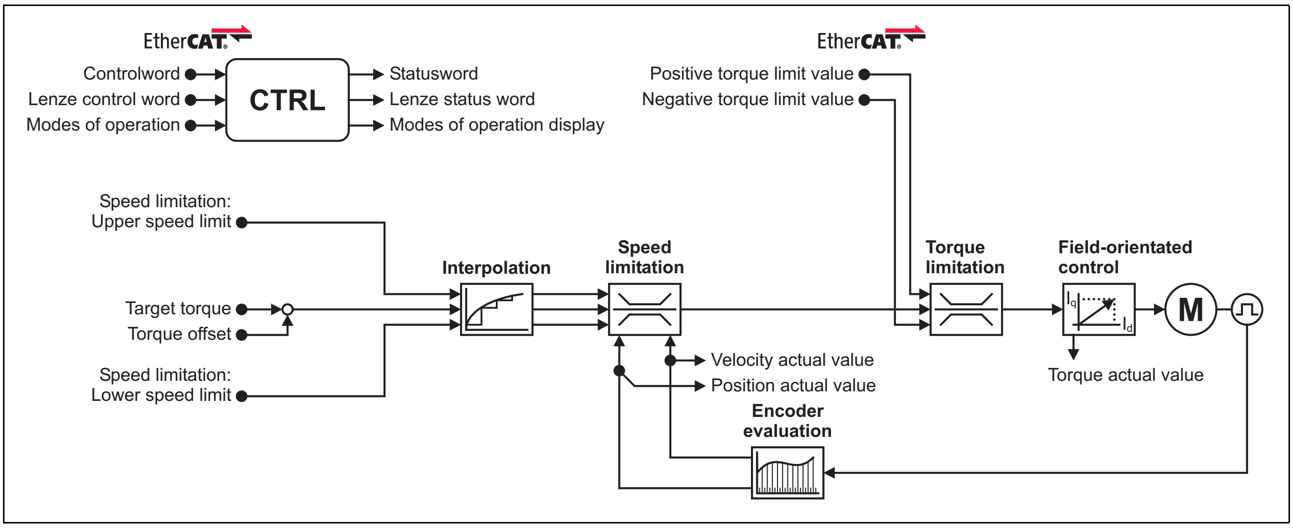
Display area (min. value | unit | max. value)			Initialisation	
0.000	V	4294967.295		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/1000	UNSIGNED_32

0x60B2 | 0x68B2 - Torque offset

Additive value for the target torque or torque feedforward control
- 100 % ≡ rated motor torque (0x6076 or 0x6876 for axis B)

Setting range (min. value | unit | max. value)			Lenze setting	
-3276.8	%	3276.7	0.0 %	
☑ Write access ☐ CINH ☑ OSC ☑ P ☑ RX ☐ TX			Scaling: 1/10	INTEGER_16

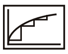
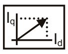
7.10.3 Signal flow



[7-8] Signal flow of the servo control in "Cyclic sync torque mode" (simplified representation)

Short overview of the most important parameters:

Function	Object		Name
	Axis A	Axis B	
Input data ●—	0x6040	0x6840	Controlword
	0x2830	0x3030	Lenze control word
	0x6060	0x6860	Operating mode: Selection
	0x2946:1	0x3146:1	Speed limitation: Upper speed limit
	0x60B2	0x68B2	Torque offset
	0x6071	0x6871	Target torque
	0x2946:2	0x3146:2	Speed limitation: Lower speed limit
	0x60E0	0x68E0	Positive torque limit value
	0x60E1	0x68E1	Negative torque limit value
Output data —▸	0x6041	0x6841	Statusword
	0x2831	0x3031	Lenze status word
	0x6061	0x6861	Modes of operation display
	0x6064	0x6864	Position actual value
	0x606C	0x686C	Velocity actual value
	0x6077	0x6877	Torque actual value

Function	Object		Name
	Axis A	Axis B	
Interpolation 	0x60C2	0x68C2	Interpolation time period
Speed limitation 	0x6080	0x6880	Max motor speed
	0x2903	0x3103	Speed: Speed setpoint - filter time
Torque limitation 	0x60E0	0x68E0	Positive torque limit value
	0x60E1	0x68E1	Negative torque limit value
	0x6076	0x6876	Motor rated torque
	0x6072	0x6872	Max torque
	0x2944:1	0x3144:1	Notch filter 1: Frequency
	0x2944:2	0x3144:2	Notch filter 1: Bandwidth
	0x2944:3	0x3144:3	Notch filter 1: Damping
	0x2944:4	0x3144:4	Notch filter 2: Frequency
	0x2944:5	0x3144:5	Notch filter 2: Bandwidth
	0x2944:6	0x3144:6	Notch filter 2: Damping
Field-oriented control 	0x6073	0x6873	Max current
	0x6075	0x6875	Motor rated current
	0x2941	0x3141	Current controller: Feedforward control
	0x2942:1	0x3142:1	Current controller: Gain
	0x2942:2	0x3142:2	Current controller: Reset time
	0x29E2	0x31E2	DC link circuit voltage: Filter time
	0x29E3	0x31E3	Motor voltage act. value: Filter time
	0x29E0:1	0x31E0:1	Field weakening controller: Gain
	0x29E0:2	0x31E0:2	Field weakening controller: Reset time
	0x29E1	0x31E1	Field set value limitation
	0x29C0:1	0x31C0:1	Field controller: Gain
	0x29C0:2	0x31C0:2	Field controller: Reset time
	0x2939	0x3139	Switching frequency

Tip!

A detailed representation of the signal flow with all relevant parameters can be found in the »PLC Designer« on the **Signal flow** tab for the i700 servo inverter.

7.10.4 Control commands & status information

The following control commands can be executed in "Cyclic synchronous torque mode" via the Controlword (0x6040 or 0x6840 for axis B):

Control word	Status	Function
Bit 4	0	*Reserved* (bit must be set to "0")
Bit 5	0	*Reserved* (bit must be set to "0")
Bit 6	0	*Reserved* (bit must be set to "0")
Bit 8	0 ↗ 1	Stop

The following status information is output in "Cyclic synchronous torque mode" via the Statusword (0x6041 or 0x6841 for axis B):

Status word	Status	Meaning
Bit 12	0	Cyclic sync torque mode inactive
	1	Cyclic sync torque mode active

7.11 Touch probe (TP)

A "Touch probe" (short: "TP") is an event which can for instance be actuated in an edge-controlled manner via a digital input to detect an actual value (that changes quickly) at the time of activation and to process it further within the program afterwards.

- Typical touch probe applications:
  - Homing
  - Mark synchronisation
  - Measurements of lengths
- Up to 2 touch probe channels can be used in parallel for each axis.
- Possible touch probe sources:
  - TP1 : Zero pulse position encoder or digital input DI1
  - TP2 : Zero pulse position encoder or digital input DI2

Note!

The digital inputs DI1 and DI2 can be evaluated additionally as "normal" digital inputs via the 0x60FD object any time (or 0x68FD for axis B).

Objects described in other chapters:

Object Axis A	 Axis B	Name	Data type
▶ General CiA402 parameters			
0x60FD	0x68FD	Digital inputs	UNSIGNED_32
Greyed out = read access only			

Objects described in this chapter:

Object Axis A	 Axis B	Name	Data type
Lenze-specific objects			
0x2500		Touch probe (TP): Filter time	UNSIGNED_16
0x2D00	0x3500	Touch probe (TP): Delay time	RECORD
0x2D01	0x3501	Touch probe (TP): Time stamp	RECORD
Objects according to CiA402 specification (version 3)			
0x60B8	0x68B8	Touch probe function	UNSIGNED_16
0x60B9	0x68B9	Touch probe status	UNSIGNED_16
0x60BA	0x68BA	Touch probe pos1 pos value	INTEGER_32
0x60BB	0x68BB	Touch probe pos1 neg value	INTEGER_32
0x60BC	0x68BC	Touch probe pos2 pos value	INTEGER_32
0x60BD	0x68BD	Touch probe pos2 neg value	INTEGER_32
Greyed out = read access only			

7.11.1 Default mapping

The default mapping for a touch probe detection is specified in the following objects:

Object Axis A	Object Axis B	Name	Data type
0x1604	0x1614	RPDO-->Axis A/B: Touch probe (TP)	RECORD
0x1A04	0x1A14	Axis A/B-->TPDO: Touch probe (TP)	RECORD

Date received from the controller (RPDO)

Object Axis A	Object Axis B	Name	Data type
0x60B8	0x68B8	Touch probe function	UNSIGNED_16

Data transmitted to the controller (TPDO)

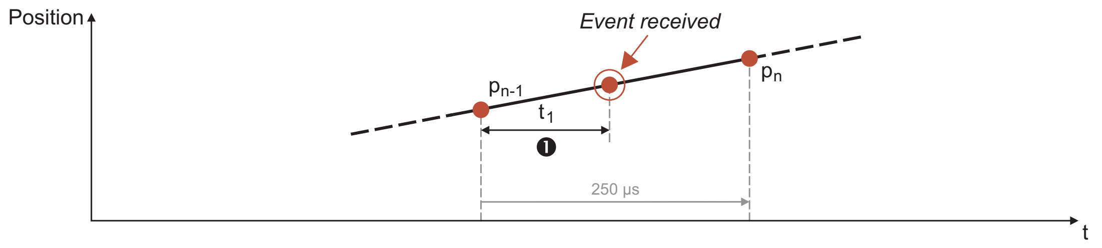
Object Axis A	Object Axis B	Name	Data type
0x60B9	0x68B9	Touch probe status	UNSIGNED_16
0x60BA	0x68BA	Touch probe pos1 pos value	INTEGER_32
0x60BB	0x68BB	Touch probe pos1 neg value	INTEGER_32
0x60BC	0x68BC	Touch probe pos2 pos value	INTEGER_32
0x60BD	0x68BD	Touch probe pos2 neg value	INTEGER_32

7.11.2 General functional principle

If an event occurs at the touch probe source configured, a time stamp is recorded in the i700 servo inverter.

The time stamp recorded is related to the system time and can thus be separated into two parts: One part is the control cycle within which the event has occurred. The other part is the time difference between the start of the control cycle determined and the actual detection of the event.

A history buffer enables the i700 servo inverter to know the last n position values. Thus, the actual position at the start and end of the control cycle within which the event has occurred is known. Linear interpolation is executed between these two position grid points. The result is the exact position at the motor shaft at the time the event is triggered:



P_{n-1} Actual position grid point 1

P_n Actual position grid point 2

❶ Time difference between the start of the control cycle determined and the actual detection of the event

[7-9] Determination of the exact position by linear interpolation (principle)

The position grid points are detected in the i700 servo inverter in a grid of 250 µs. After a touch probe is tripped, the input is deactivated for up to 250 µs in order to avoid bouncing. Thus the maximum frequency for the tripping of touch probes is 4 kHz.

If, in contrast to the steady motion outlined in figure [7-9], an accelerated motion is taken as a basis, the 250 µs grid also allows for a very good linear position reconstruction, since the velocity change at the motor shaft can only have a minor effect during 250 µs.

7.11.3 Filtering the touch probe signal

A common filter time (debounce time) can be parameterised for the touch probe inputs in order to debounce the TP signals so that there is no response to external interference signals.

- Every 31 µs, the signal status at the TP input is detected for the debounce filter and a new value is assigned to the filter.
- Via the following object, the filter time for all touch probe inputs of both axes of the device is set. A separate setting for a touch probe or an axis is not possible.

0x2500 - Touch probe (TP): Debounce time

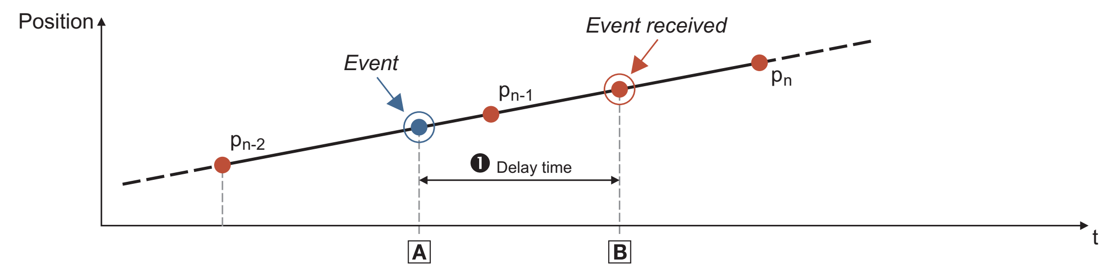
Note: Since the filter is scanned with 32 kHz, there are discrete, adjustable values. After entering an optional filter time between 0 and 1984 µs, the value is automatically internally rounded down to the next adjustable value and is also displayed on read request.
- The filter time is automatically taken into consideration in the TP calculation.
- If the setting is "0", the filter is deactivated.

Setting range (min. value | unit | max. value)			Lenze setting	
0	us	1984	0 us	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

7.11.4 Compensation of runtime delays

In reality, both the input connection in the i700 servo inverter and the touch probe sensor have runtime delays (latencies) themselves. They can be taken into consideration for calculating the actual tripping time and thus the actual position at the time of tripping.

In the following illustration, the event is detected at the time B in the i700 servo inverter. Due to the input connection and the sensor used, however, the signal has undergone a runtime delay; the actual physical event has already occurred at the time A. In order to compensate this runtime delay, a corresponding delay time which is taken into consideration for the determination of the control cycle and interpolation of the position (see figure [7-9]) can be set for each touch probe channel.



A Actual physical event
B Electrical recognition of the event in the i700 servo inverter
1 Delay time between the actual physical event and the electrical recognition

[7-10] Compensation of runtime delays (principle)

Delay times of the digital inputs and required minimum signal lengths

The table below lists the delay times typical for the digital inputs of the i700 servo inverter and the required minimum signal lengths:

Digital signal	Typical delay time	Minimum signal length
Rising edge (HIGH pulse)	5 - 7 µs	7 µs
Falling edge (LOW pulse)	approx. 40 µs	50 µs

0x2D00 | 0x3500 - Touch probe (TP): Dead time compensation

Delay time for the compensation of runtime delays (latencies)
- Set the sum of individual delays for each touch probe channel here (sensor, cable, and input connection).

Sub.	Name	Lenze setting	Data type
1	Dead time compensation: TP1 dead time	0.000 ms	UNSIGNED_16
2	Dead time compensation: TP2 dead time	0.000 ms	UNSIGNED_16

Subindex 1: **Dead time compensation: TP1 delay time**				
Setting range (min. value | unit | max. value)			Lenze setting	
0.000	ms	7.000	0.000 ms	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/1000	UNSIGNED_16

Subindex 2: **Dead time compensation: TP2 delay time**				
Setting range (min. value | unit | max. value)			Lenze setting	
0.000	ms	7.000	0.000 ms	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/1000	UNSIGNED_16

7.11.5 Touch probe function

0x60B8 | 0x68B8 - Touch probe function

Control word for the configuration of the touch probe functionality

Setting range (min. value | unit | max. value)			Lenze setting
0		65535	0
Value is bit-coded: (☑ = bit set)			Info
Bit 0 ☐	Enable touch probe 1		0: Switch off touch probe 1 1: Enable touch probe 1
Bit 1 ☐	Trigger on first event (0) or continous (1)		Event for touch probe channel 1: 0: Record first event only 1: Record all events
Bit 2 ☐	TP: Source - TP input (0) or position encoder zero pulse (1)		Source for touch probe channel 1: 0: Digital input 1 1: Zero pulse position encoder
Bit 3 ☐	Reserved		
Bit 4 ☐	TP1: Activate sampling - rising edge		0: Deactivate sampling 1: Activate sampling
Bit 5 ☐	TP1: Activate sampling - falling edge		0: Deactivate sampling 1: Activate sampling
Bit 6 ☐	Reserved		
Bit 7 ☐	Reserved		
Bit 8 ☐	Enable touch probe 2		0: Switch off touch probe 2 1: Enable touch probe 2
Bit 9 ☐	Trigger on first event (0) or continous (1)		Event for touch probe channel 2: 0: Record first event only 1: Record all events
Bit 10 ☐	TP: Source - TP input (0) or position encoder zero pulse (1)		Source for touch probe channel 2: 0: Digital input 2 1: Zero pulse position encoder
Bit 11 ☐	Reserved		
Bit 12 ☐	TP2: Activate sampling - rising edge		0: Deactivate sampling 1: Activate sampling
Bit 13 ☐	TP2: Activate sampling - falling edge		0: Deactivate sampling 1: Activate sampling
Bit 14 ☐	Reserved		
Bit 15 ☐	Reserved		
☑ Write access ☐ CINH ☑ OSC ☑ P ☑ RX ☐ TX			UNSIGNED_16

7.11.6 Touch probe status

0x60B9 | 0x68B9 - Touch probe status

Status of the touch probe functionality

Display area (min. value | unit | max. value)			Initialisation	
0		65535	0	
Value is bit-coded:			Info	
Bit 0	TP1: Active		0: Touch probe channel 1 deactivated 1: Touch probe channel 1 activated	
Bit 1	TP1: Position detected - rising edge		0: Position not detected 1: Position detected	
Bit 2	TP1: Position detected - falling edge		0: Position not detected 1: Position detected	
Bit 3	Reserved			
Bit 4	Reserved			
Bit 5	Reserved			
Bit 6	Touch probe (TP): Level at release time		Level at detection via touch probe channel 1: 0: LOW level 1: HIGH level	
Bit 7	Reserved			
Bit 8	TP2: Active		0: Touch probe channel 2 deactivated 1: Touch probe channel 2 activated	
Bit 9	TP2: Position detected - rising edge		0: Position not detected 1: Position detected	
Bit 10	TP2: Position detected - falling edge		0: Position not detected 1: Position detected	
Bit 11	Reserved			
Bit 12	Reserved			
Bit 13	Reserved			
Bit 14	Touch probe (TP): Level at release time		Level at detection via touch probe channel 2: 0: LOW level 1: HIGH level	
Bit 15	Reserved			
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX				UNSIGNED_16

7.11.7 Time stamps and positions detected

Note!

In the event of a "Continuous trigger" touch probe configuration, a newly detected value overwrites the value detected before, even if it has not been retrieved by the controller.

0x2D01 | 0x3501 - Touch probe (TP): Time stamp

Time stamp for the positions detected via touch probe
- From version 01.05 on, the time stamps of the TPs are coupled to the Ethercat DC time and directly map the lower 32 bits of the DC time. Thus, an absolute time of the TP event is available for use in the system.

Sub.	Name	Lenze setting	Data type
1	TP1: Time stamp - rising edge	0 ns	UNSIGNED_32
2	TP1: Time stamp - falling edge	0 ns	UNSIGNED_32
3	TP2: Time stamp - rising edge	0 ns	UNSIGNED_32
4	TP2: Time stamp - falling edge	0 ns	UNSIGNED_32
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			

0x60BA | 0x68BA - Touch probe pos1 pos value

Touch probe position 1 detected with a rising edge

Display area (min. value | unit | max. value)			Initialisation	
-2147483648	[Pos unit]	2147483647	0 [Pos unit]	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX				INTEGER_32

0x60BB | 0x68BB - Touch probe pos1 neg value

Touch probe position 1 detected with a falling edge

Display area (min. value | unit | max. value)			Initialisation	
-2147483648	[Pos unit]	2147483647	0 [Pos unit]	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX				INTEGER_32

0x60BC | 0x68BC - Touch probe pos2 pos value

Touch probe position 2 detected with a rising edge

Display area (min. value | unit | max. value)			Initialisation	
-2147483648	[Pos unit]	2147483647	0 [Pos unit]	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX				INTEGER_32

0x60BD | 0x68BD - Touch probe pos2 neg value

Touch probe position 2 detected with a falling edge

Display area (min. value | unit | max. value)			Initialisation	
-2147483648	[Pos unit]	2147483647	0 [Pos unit]	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX				INTEGER_32

8 Monitoring functions

The i700 servo inverter is provided with various monitoring functions, which are described in the following subchapters.

Objects described in this chapter

Object Axis A	 Axis B	Name	Data type
0x2D40	0x3540	Ixt utilisation	RECORD
0x2D44	0x3544	Motor speed monitoring	RECORD
0x2D45	0x3545	Motor phase failure detection	RECORD
0x2D46	0x3546	Monitoring: Ultimate motor current	RECORD
0x2D49	0x3549	Motor temperature monitoring: Parameter	RECORD
0x2D4C	0x354C	Motor utilisation (I^2xt): Parameter for the thermal model	RECORD
0x2D4D	0x354D	Motor utilisation (I^2xt): User-definable characteristic	RECORD
0x2D4E	0x354E	Motor utilisation (I^2xt): Motor overload warning threshold	UNSIGNED_16
0x2D4F	0x354F	Motor utilisation (I^2xt): Actual utilisation	UNSIGNED_32
0x2D50	0x3550	Motor utilisation (I^2xt): Motor overload error	UNSIGNED_32
0x2D84	0x3584	Heatsink temperature	RECORD

8.1 24-V supply voltage monitoring

The 24 V supply voltage is monitored by means of two specified voltage thresholds.

- The following table describes these thresholds and shows the device response resulting from the level of the supply voltage:

Threshold	Voltage	Device response	
U24 warn	21.45 V ±5 %	Above threshold:	Normal operation
		Below threshold:	Normal operation with warning
U24 down	18.15 V ±5 %	Above threshold:	Normal operation with warning
		Below threshold:	No operation • Response with regard to the EtherCAT master and adjacent EtherCAT slaves is undefined.

Related topics:

▸ Monitoring of the DC-bus voltage (📖 48)

8.2 Monitoring of the power section and device utilisation (Ixt)

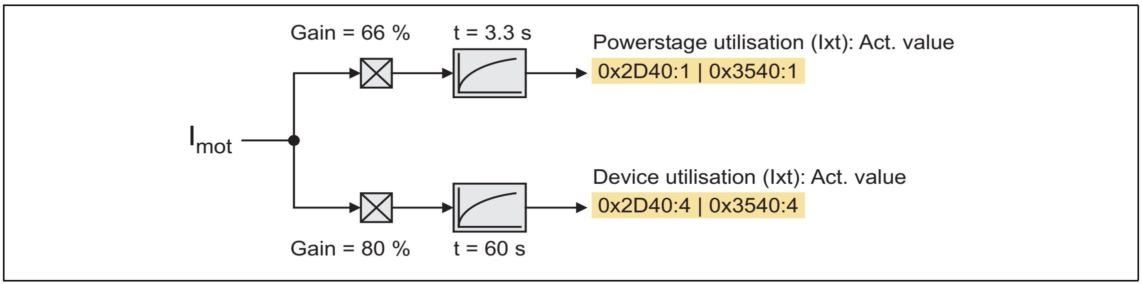
The monitoring of the device utilisation primarily protects the power section. Indirectly, also other components such as conductors and terminals are protected. The monitoring system is designed as simply as possible, in order to facilitate the determination of the reserves until disconnection, and to carry out the drive dimensioning accordingly. Only cycles the arithmetic mean (AV) of which does not exceed the continuous device current are permissible.

"i700 servo inverter" hardware manual

In order to determine the reserves until Ixt disconnection, the rated data depending on the switching frequency and mains operation such as the rated device current and maximum current for the respective i700 design can be gathered from the hardware manual. (See chapter 4.2.4 "Axis modules" and chapter 4.3 "Overcurrent operation".)

The motor current is evaluated depending on the continuously permissible rated current at the effective switching frequency and mains voltage and is provided to two PT1 timing elements:

- The PT1 timing element with a time constant of t = 3.3 s simulates the thermal response of the power section (power semiconductor).
- The PT1 timing element with a time constant of t = 60 s serves to monitor the cycle with maximum current in order to protect the device from thermal overload (heatsink temperature).



[8-1] Ixt monitoring signal flow

The outputs of the two PT1 timing elements show the current utilization of power section and the entire device.

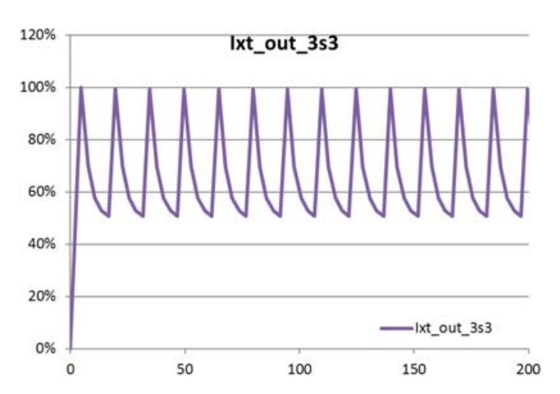
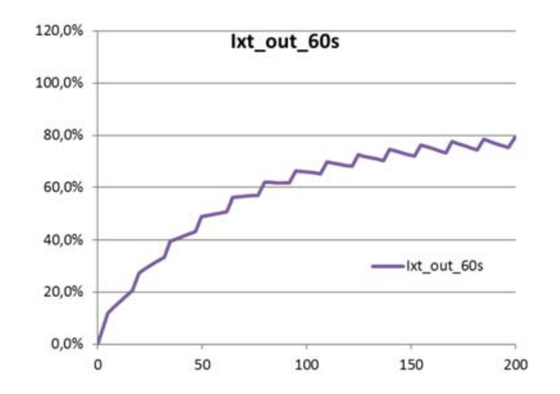
Gain factors and filter time constants are defined such that within a certain cycle time a specific overload current can be active for a certain period with subsequent recovery phase (75 % rated current).

The following load change cycles are defined for the i700 servo inverter:

- 15-second cycle:
  200 % rated current for 3 s, recovery phase 12 s with 75 % rated current
- 24-second cycle:
  175 % rated current for 6 s, recovery phase 18 s with 75 % rated current
- 3-minute cycle:
  150 % rated current for 60 s, recovery phase 120 s with 75 % rated current

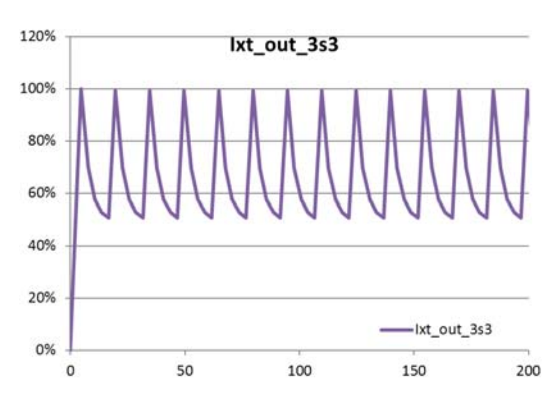
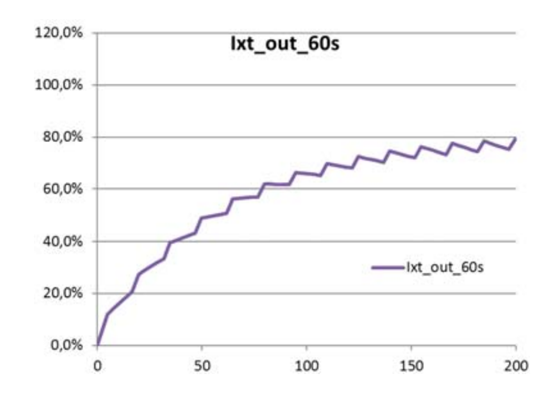
1. Load cycle

15-second cycle with a 200 %/3 s overload phase and a 75 %/12 s recovery phase.

PT1 with 3.3 s filter time constant for power section protection	PT1 with 60 s filter time constant for device protection
	

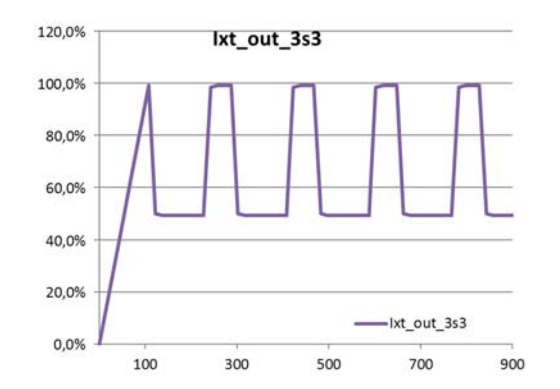
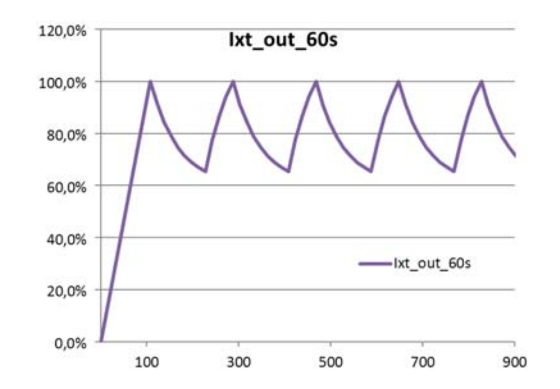
2. Load cycle

24-second cycle with a 175 %/6 s overload phase and a 75 %/18 s recovery phase.

PT1 with 3.3 s filter time constant for power section protection	PT1 with 60 s filter time constant for device protection
	

3. Load cycle

3 minute cycle with a 150 %/60 s overload phase and a 75 %/120 s recovery phase.

PT1 with 3.3 s filter time constant for power section protection	PT1 with 60 s filter time constant for device protection
	

Note!

Though continuous operation over 100 % is allowed for by the Ixt monitoring function, this is nevertheless an impermissible working point!

Parameterisation and evaluation of the monitoring function

The Ixt monitoring function can be parameterised and evaluated via the following indices. Here, a warning limit and an error limit can be set for each PT1 timing element.

- If the warning threshold (Lenze setting: 95 %) is exceeded, a warning is output in order that the master is still able to respond if required.
- If the error threshold is exceeded (Lenze setting: 101 %), the inverter is switched off for protecting the device and generates an error message.

0x2D40 | 0x3540 - Ixt utilisation

Sub.	Name	Lenze setting	Data type
▸ 1	Power section utilisation (Ixt): Actual utilisation	0 %	UNSIGNED_16
▸ 2	Power stage utilisation (Ixt): Warning threshold	95 %	UNSIGNED_16
▸ 3	Power stage utilisation (Ixt): Error threshold	101 %	UNSIGNED_16
▸ 4	Device utilisation (Ixt): Actual utilisation	0 %	UNSIGNED_16
▸ 5	Device utilisation (Ixt): Warning threshold	95 %	UNSIGNED_16
▸ 6	Device utilisation (Ixt): Error threshold	101 %	UNSIGNED_16

Subindex 1: **Power stage utilisation (Ixt): Actual utilisation**				
Display area (min. value | unit | max. value)			Initialisation	
0	%	101	0 %	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX				UNSIGNED_16

Subindex 2: **Power stage utilisation (Ixt): Warning threshold**				
Setting range (min. value | unit | max. value)			Lenze setting	
0	%	101	95 %	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

Subindex 3: **Power stage utilisation (Ixt): Error threshold**				
Display area (min. value | unit | max. value)			Initialisation	
0	%	101	101 %	
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX				UNSIGNED_16

Subindex 4: **Device utilisation (Ixt): Actual utilisation**				
Display area (min. value | unit | max. value)			Initialisation	
0	%	101	0 %	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX				UNSIGNED_16

Subindex 5: **Device utilisation (Ixt): Warning threshold**				
Setting range (min. value | unit | max. value)			Lenze setting	
0	%	101	95 %	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

Subindex 6: **Device utilisation (Ixt): Error threshold**				
Display area (min. value | unit | max. value)			Initialisation	
0	%	101	101 %	
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX				UNSIGNED_16

8.3 Monitoring of the heatsink temperature

In order to avoid impermissible heating of the device, the heatsink temperature is detected and monitored.

Note!

In the temperature range 0 ... 80 °C, the heatsink temperature is measured with a tolerance of -2 ... +4 °C. Outside this temperature range, the measuring accuracy strongly decreases.

0x2D84 | 0x3584 - Heatsink temperature

Sub.	Name	Lenze setting	Data type
1	Heatsink temperature: Actual temperature		INTEGER_16
2	Heatsink temperature: Warning threshold	80.0 °C	INTEGER_16
3	Heatsink temperature: Threshold - switch-on fan	55.0 °C	INTEGER_16
4	Heatsink temperature: Threshold - switch-off fan	45.0 °C	INTEGER_16

Subindex 1: **Heatsink temperature: Actual temperature**				
Display of the current heatsink temperature				
Display area (min. value | unit | max. value)			Initialisation	
-3276.8	°C	3276.7		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☐ TX			Scaling: 1/10	INTEGER_16

Subindex 2: **Heatsink temperature: Warning threshold**				
Warning threshold for temperature monitoring • If the heatsink temperature exceeds the threshold value set here, the device reports a warning. The warning message is reset with a hysteresis of approx. 5 °C. • If the heatsink temperature increases further and exceeds the non-adjustable error threshold (100 °C), the device changes to the "Fault" device status. Further operation of the i700 servo inverter is disabled by controller inhibit.				
Setting range (min. value | unit | max. value)			Lenze setting	
50.0	°C	100.0	80.0 °C	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	INTEGER_16

Subindex 3: **Heatsink temperature: Threshold for fan switch-on**				
Switch-on threshold for device fans • If the heatsink temperature exceeds the threshold value set here, the device fan is switched on. This also happens if the switch-off threshold is parameterised higher than the switch-on threshold by mistake. • In order to activate the fan, the switch-on and switch-off threshold can be set to the maximum value as then the device simultaneously switches to the "Fault" status. • Bigger designs of the Servo-Inverter i700 are provided with an internal fan. This is triggered at the same time as the heatsink fan and does not have internal switch-on or switch-off thresholds. **Note!** Increasing the switch-on threshold can cause higher device temperatures. This may reduce the service life of the device!				
Setting range (min. value | unit | max. value)			Lenze setting	
0.0	°C	100.0	55.0 °C	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	INTEGER_16

Subindex 4: **Heatsink temperature: Threshold for fan switch-off**				
Switch-off threshold for device fans • If the heatsink temperature falls below the threshold value set here, the device fan is switched off. This only happens if the switch-off threshold is parameterised lower than the switch-on threshold. • Bigger designs of the Servo-Inverter i700 are provided with an internal fan. This is triggered at the same time as the heatsink fan and does not have internal switch-on or switch-off thresholds.				
Setting range (min. value | unit | max. value)			Lenze setting	
0.0	°C	100.0	45.0 °C	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	INTEGER_16

8.4 Monitoring of the motor utilisation (I²xt)

This monitoring detects the thermal utilisation of the motor by calculating the thermal motor utilisation from the detected motor currents based on a mathematical model and displaying it in the object 0x2D4F (or 0x354F for axis B).

In case of permanent overload and excess of the warning threshold set in the object 0x2D4E (or 0x354E for axis B), a warning is output in order that the higher-level Controller is still able to respond and reduce the motor load or interrupt the operation.

From version 01.04 on, an error response in the object 0x2D50:1 (or 0x3550:1 for axis B) can be parameterised as well if the disconnection should not or cannot be executed by a higher-level Controller.

> STOP
>
> **Stop!**
>
> Monitoring the motor utilisation (I²xt) is not a means for full motor protection!
>
> Since the motor utilisation calculated in the thermal model gets lost after mains switching, the following operating states cannot be determined correctly:
> - Restarting (after mains switching) of a motor that is already very hot.
> - Change of the cooling conditions (e.g. cooling air flow interrupted or too warm).
>
> Full motor protection requires additional measures such as the evaluation of temperature sensors that are located directly in the winding or the use of thermal contacts.

During the calculation, the speed dependence of the permissible motor load and thus of the permissible current (difference between the standstill current and rated current is taken into consideration. This is done via object 0x2D4D (or 0x354D for axis B).

The threshold value for the warning output can be adapted via the following object:

0x2D4E | 0x354E - Motor utilisation (I²xt): Motor overload warning threshold

Setting range (min. value | unit | max. value)			Lenze setting	
0	%	250	100 %	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

The current thermal motor utilisation is shown in the following object:

0x2D4F | 0x354F - Motor utilisation (I²xt): Actual utilisation

Display area (min. value | unit | max. value)			Initialisation	
0	%	250		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX				UNSIGNED_16

0x2D50 | 0x3550 - motor utilisation (I²xt): Motor overload error

From version 01.04 onwards
If the disconnection must not or cannot be executed by a higher-level Controller, the operation can be interrupted by the following parameter setting:
- As a response to the excess of the adjustable error threshold, set "1: Fault" in subindex 1.
- In subindex 2, set the threshold value as error threshold which in case of being exceeded triggers an interruption of the operation.

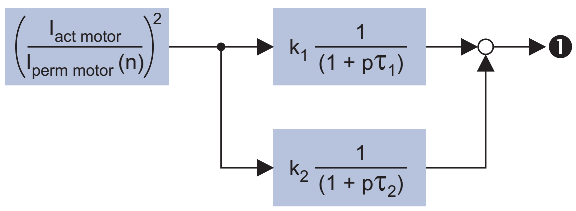
Sub.	Name	Lenze setting	Data type
▸ 1	Motor utilisation (I²xt): Response	0: No Response	UNSIGNED_8
▸ 2	Motor utilisation (I²xt): Error threshold	105 %	UNSIGNED_16

Subindex 1: **Motor utilisation (I²xt): Response**			
Selection list (Lenze setting printed in bold)			
0	**No response**		
1	Fault		
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_8

Subindex 2: **Motor utilisation (I²xt): Error threshold**				
Setting range (min. value | unit | max. value)			Lenze setting	
0	%	250	105 %	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

The introduction of a two-component model with two time constants (one for the winding and the housing/laminated core, respectively) serves to display the thermal behaviour of the motors up to 500% of the rated current.

Structure of the I^2xt monitoring



❶ Thermal utilisation of the motor in [%]

Axis A	Axis B	Symbol	Description	Dimension unit
-	-	$I_{act\ motor}$	Actual motor current	A
-	-	$I_{perm\ motor}$	Permissible motor current (speed-dependent)	A
0x2D4C:1	0x354C:1	τ_1	Therm. time constant coil	s
0x2D4C:2	0x354C:2	τ_2	Therm. time constant of the laminated core	s
0x2D4C:3	0x354C:3	k_1	Percentage of the winding in the final temperature	%
-	-	k_2	Percentage of the laminated core in the final temperature: k_2 = 100 % - k_1	%

0x2D4C | 0x354C - Motor utilisation (I²xt): Parameter for the thermal model

Sub.	Name	Lenze setting	Data type
▸ 1	Motor utilisation (I²xt): Thermal time constant - winding	60 s	UNSIGNED_16
▸ 2	Motor utilisation (I²xt): Thermal time constant - laminations	852 s	UNSIGNED_16
▸ 3	Motor utilisation (I²xt): Influence of the winding	27 %	UNSIGNED_8
▸ 4	Motor utilisation (I²xt): Starting value	0 %	UNSIGNED_16

Subindex 1: **Motor utilisation (I²xt): Thermal time constant of the winding**				
Setting range (min. value | unit | max. value)			Lenze setting	
1	s	36000	60 s	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

Subindex 2: **Motor utilisation (I²xt): Thermal time constant - laminated core**				
Setting range (min. value | unit | max. value)			Lenze setting	
1	s	36000	852 s	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

Subindex 3: **Motor utilisation (I²xt): Influence of the winding**				
Setting range (min. value | unit | max. value)			Lenze setting	
0	%	100	27 %	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_8

Subindex 4: **Motor utilisation (I²xt): Starting value**				
Setting range (min. value | unit | max. value)			Lenze setting	
0	%	250	0 %	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

Calculation with only one time constant

If k1 = "0 %" is set, the part of the winding is not taken into consideration and the thermal model is only calculated using the time constant set for the housing/laminated core. This setting is e.g. required if only the time constant of the laminated core (T2) is known.

Parameter setting of the time constant and the influence of the winding on motors of other manufacturers

When the influence of the winding is activated, the i²xt monitoring becomes more sensible as if only the influence of the laminated core would be used for monitoring purposes.

The necessity to activate the influence of the winding rises with the increasing utilisation of the motor overload capacity. It also rises with applications where the motor is at standstill for longer periods or cyclically and a load ≥ permanent standstill current is applied.

For determining the values for the thermal time constant, try to get the data from the motor manufacturer. If this is not possible, you can use the data of a comparable Lenze motor.

Conditions for comparability are similar values in case of the following motor features:

- Square dimensions of the motor (active part)
- Length of the active part (if available)
- Permanent standstill current Io [A_RMS]
- Peak current/overload capacity [A_RMS]
- Copper resistance of the winding at 20 °C [Rphase]

Example:

Motor features	Data of the third-party motor	Data of a comparable Lenze motor (from motor catalogue)
Square dimension	95 mm	MCS09xxx = 89 mm
Standstill current	2.2 A	MCS09F38 = 3.0 A
Peak current	7.3 A	MCS09F38 = 15 A
Phase resistance	5.1 Ohm	MCS09F38 = 5.2 Ohm

When the "MCS09F38" Lenze motor is selected from the motor catalogue of the »PLC Designer«m the following values are displayed:

- Thermal time constant of the winding = 126 s ➜ 0x2D4C:1 (or 0x354C:1 for axis B)
- Influence of the winding = 27 % ➜ 0x2D4C:3 (or 0x354C:3 for axis B)

Speed-dependent evaluation of the motor current

By selecting a characteristic, the permissible motor current is evaluated depending on speed for calculating the thermal motor utilisation. For this purpose, up to four operating points on the S1 characteristic of a motor can be used.

- The S1 characteristic can be found in the technical data sheet/catalogue of the respective motor.
- The representation in the objects /characteristic is carried out as relative values with reference to rated values.

Note!

When you select a Lenze motor from the catalogue and transfer its parameters into the controller, a typical characteristic is automatically set for the selected motor. A deviating parameterisation is only required if the motor is operated in ambient conditions which demand a general derating. Example: Use in site altitudes > 1000 m.

In case of motors of other manufacturers, the operating points have to be parameterised based on the data sheet information.

Operating points		
❶	Standstill n01-I01	For motors, this operating point is often described with the no-Io values.
❷	Reference point n02-I02	If the value falls below the speed n02, a derating in the current is required because: • the motor cooling of self-ventilated motors deteriorates considerably. • a DC current load causes an increased power loss in a winding. For motors, this operating point is also described with the no*-Io* values.
❸	Rated point (n03=nrated)-(I03=Irated)	Rated values of the motor are the reference for all operating points of the i^2xt monitoring.
❹	Field weakening n04-I04	This operating point should be parameterised independently of the use in the current application.

0x2D4D | 0x354D - Motor utilisation (I^2xt): User-definable characteristic

Sub.	Name	Lenze setting	Data type
1	I^2xt: x1 = n01/nN (n01 ~ 0)	0 %	UNSIGNED_16
2	I^2xt: y1 = i01/iN (x = n01 ~ 0)	100 %	UNSIGNED_16
3	I^2xt: x2 = n02/nN (n02 = limit - reduced cooling)	0 %	UNSIGNED_16
4	I^2xt: y2 = i02/iN (x = n02 = limit - reduced cooling)	100 %	UNSIGNED_16
5	I^2xt: x3 = n03/nN (n03 = rated speed)	100 %	UNSIGNED_16
6	I^2xt: y3 = i03/iN (x = n03 = rated speed)	100 %	UNSIGNED_16
7	I^2xt: x4 = n04/nN (n04 = limit - field weakening)	100 %	UNSIGNED_16
8	I^2xt: y4 = i04/iN (x = n04 = limit - field weakening)	100 %	UNSIGNED_16
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			

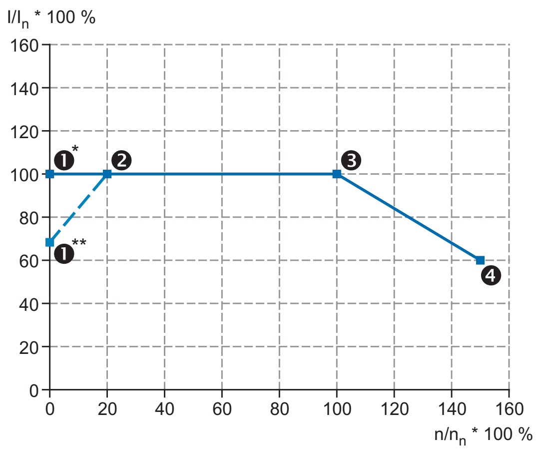
An example of how to enter the characteristic can be found in the following subchapter.

8.4.1 Example of how to enter the characteristic for standard and servo motors

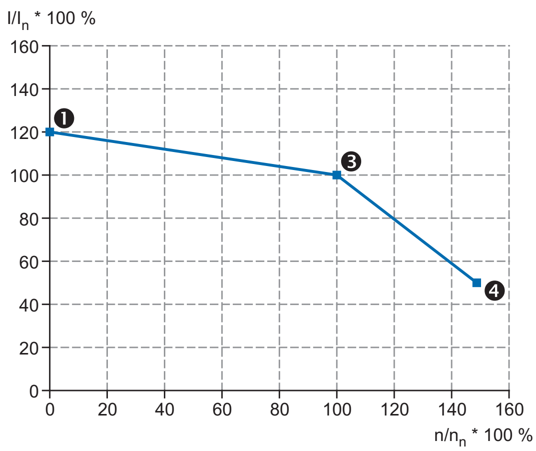
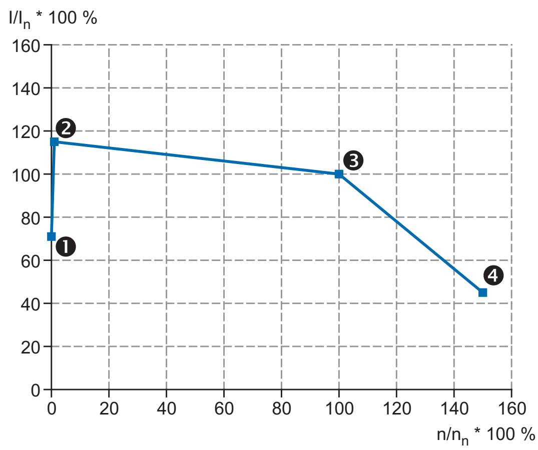
The required data of the operating points result from the S1 characteristic of the prevailing motor:

Examples for S1 characteristics

Standard motor



* Standard motor forced ventilated
** Standard motor self-ventilated

Servo motor	Servo motor with derating at standstill
	

Object Axis A	 Axis B	Characteristic point		Info
0x2D4D:1	0x354D:1	❶	n_1/n_n * 100 %	Speed = "0" (standstill)
0x2D4D:2	0x354D:2		I_1/I_n * 100 %	Permissible motor current at standstill
0x2D4D:3	0x354D:3	❷	n_2/n_n * 100 %	Speed from which the current must be reduced for self-ventilated motors. • Below this speed the cooling air flow of the integral fan is not sufficient anymore.
0x2D4D:4	0x354D:4		I_2/I_n * 100 %	Permissible motor current at speed n_2 (torque reduction)
0x2D4D:5	0x354D:5	❸	n_3/n_n * 100 %	Rated speed
0x2D4D:6	0x354D:6		I_3/I_n * 100 %	Permissible motor current at rated speed
0x2D4D:7	0x354D:7	❹	n_4/n_n * 100 %	Speed above the rated speed (in the field weakening range for asynchronous motors)
0x2D4D:8	0x354D:8		I_4/I_n * 100 %	Permissible motor current at speed n_4 (field weakening)

The speed-dependent evaluation of the permissible motor current can actually be switched off by parameterising all 8 characteristic points to "100 %".

Stop!

Applications with standard motors

Self-ventilated standard motors are insufficiently protected at low speeds if the speed-dependent evaluation of the permissible motor current is not effective (all characteristic points = "100 %").

Applications with PM synchronous motors

Please check for every individual case which r.m.s. value can be used to permanently operate the motor at standstill.

In case of some motors, a derating I1/In < 100 % is required when n1/nn = 0 %. This serves to prevent an overload of individual motor phases as their power loss doubles with continuous DC current load. (It is called DC current load as the field frequency amounts to 0 Hz at standstill.)

8.4.2 UL 508-compliant I²xt motor overload monitoring

If the compliance with the UL 508 standard is required for the operation of the motor and the UL 508 compliant motor load monitoring is implemented by the mathematical model of the I²xt monitoring, the following conditions have to be met.

Note!

The settings described have been taken from the current UL 508 standard. If the UL 508 standard is changed, the parameter setting has to be adapted.

The i700 servo inverter only outputs one warning if the warning limit has been exceeded (0x2D4E or 0x354E for axis B). The i700 servo inverter cannot interrupt the operation of the motor. The motor load has to be interrupted by a higher-level Controller.

Stop!

From version 01.04 onwards:

If the disconnection must not or cannot be executed by a higher-level Controller, the operation has to be interrupted by the following parameter setting:

1. Set the response "1: Fault" to the excess of the adjustable error threshold in the object 0x2D50:1 (or 0x3550:1 for axis B).
2. Set the threshold value as error threshold which in case of being exceeded triggers an interruption of the operation in the object 0x2D50:2 (or 0x3550:2 for axis B).

UL 508 condition 1:

With a motor load of 600 %, the I²xt warning has to be output within 20 seconds.

- A motor load of 600 % occurs if the r.m.s. value of the total motor current displayed in the object 0x2DD1:5 (or 0x35D1:5 for axis B) corresponds to 6 times the rated motor current (0x6075 or 0x6875 for axis B).

This condition can be met by the following parameter setting:

Object		Name	Setting
Axis A	Axis B		
0x2D4C:2	0x354C:2	Motor utilisation (I²xt): Thermal time constant - laminations	≤ 709 s
0x2D4C:3	0x354C:3	Motor utilisation (I²xt): Influence of the winding	0 %
0x2D4E	0x354E	Motor utilisation (I²xt): Motor overload warning threshold	100 %
From version 01.04 onwards:			
0x2D50:1	0x3550:1	Motor utilisation (I²xt): Response	1: Fault
0x2D50:2	0x3550:2	Motor utilisation (I²xt): Error threshold	100 %

UL 508 condition 2:

With a motor load of 110 % and a motor rotating field frequency of 10 Hz, the I^2xt warning has to be output faster than with a motor rotating field frequency of 20 Hz.

- The current motor rotating field frequency is displayed in the object 0x2DDD (or 0x35DD for axis B).
- A motor load of 110 % occurs if the r.m.s. value of the total motor current displayed in the object 0x2DD1:5 (or 0x35D1:5 for axis B) corresponds to 1.1 times the rated motor current (0x6075 or 0x6875 for axis B).

This condition can be met by the following parameter setting:

Object		Name	Setting
Axis A	Axis B		
0x2D4D:1	0x354D:1	I^2xt: x1 = n01/nN (n01 ~ 0)	0 %
0x2D4D:2	0x354D:2	I^2xt: y1 = i01/iN (x = n01 ~ 0)	< 100 %
0x2D4D:3	0x354D:3	I^2xt: x2 = n02/nN (n02 = limit - reduced cooling) Setting: 100 % * 20 Hz / rated motor frequency 0x2C01:5 or 0x3401:5)	
0x2D4D:4	0x354D:4	I^2xt: y2 = i02/iN (x = n02 = limit - reduced cooling)	≥ 100 %

UL 508 condition 3:

After mains switching and a motor load > 100 %, the I^2xt warning has to be output faster than in the same load case before mains switching.

- A motor load of > 100 % occurs if the r.m.s. value of the total motor current displayed in the object 0x2DD1:5 (or 0x35D1:5 for axis B) is higher than the rated motor current (0x6075 or 0x6875 for axis B).

This condition can be met by the following parameter setting:

Object		Name	Setting
Axis A	Axis B		
0x2D4C:4	0x354C:4	Motor utilisation (I^2xt): Starting value	> 0 %

8.5 Motor temperature monitoring

0x2D49 | 0x3549 - Motor temperature monitoring: Parameters

Sub.	Name	Lenze setting	Data type
1	Motor temperature monitoring: Sensor type	0: KTY83-110	UNSIGNED_8
2	Motor temperature monitoring: Response	1: Trouble	UNSIGNED_8
3	Motor temperature monitoring: Warning threshold	145.0 °C	INTEGER_16
4	Motor temperature monitoring: Error threshold	155.0 °C	INTEGER_16
5	Motor temperature monitoring: Actual motor temperature		INTEGER_16
6	Thermal sensor characteristic: Grid point 1 - temperature	25.0 °C	INTEGER_16
7	Thermal sensor characteristic: Grid point 2 - temperature	150.0 °C	INTEGER_16
8	Spec. charact.: resistance sampling point 1	1000 Ohm	INTEGER_16
9	Spec. charact.: resistance sampling point 2	2225 Ohm	INTEGER_16

Subindex 1: **Motor temperature monitoring: Sensor type**			
Selection of the motor temperature sensor used			
Selection list (Lenze setting printed in bold)		Info	
0	**KTY83-110**		
1	KTY83-110 + 2 x PTC 150 °C (series connection)		
2	KTY84-130		
3	Spec. charact. curve	From version 01.03 Spec. charact. curve for motor temperature sensor	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_8

Subindex 2: **Motor temperature monitoring: Response**			
Response when tripping the motor temperature monitoring • In order to exclude an unwanted tripping of the fault during initial commissioning without connected motor, the error response is a warning as long as the device is in the "switch-on inhibited" status. If this status is left, the error response parameterised here is triggered.			
Selection list (Lenze setting printed in bold)			
0	No response		
1	**Fault**		
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_8

Subindex 3: **Motor temperature monitoring: Warning threshold**				
Warning threshold for motor temperature monitoring • Reset is effected with a hysteresis of 5 °C.				
Setting range (min. value | unit | max. value)			Lenze setting	
-3276.8	°C	3276.7	145.0 °C	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	INTEGER_16

Subindex 4: **Motor temperature monitoring: Error threshold**	
Error threshold for motor temperature monitoring • Reset is effected with a hysteresis of 5 °C.	
Setting range (min. value | unit | max. value)	Lenze setting

Subindex 4: **Motor temperature monitoring: Error threshold**				
-3276.8	°C	3276.7	155.0 °C	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	INTEGER_16

Subindex 5: **Motor temperature monitoring: Actual motor temperature**				
Display of the current motor temperature				
Display area (min. value | unit | max. value)			Initialisation	
-666.0	°C	666.0		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/10	INTEGER_16

Subindex 6: **Spec. charact.: temperature sampling point 1**				
From version 01.03 The special thermal sensor characteristic is selected by setting "3" in 0x2D49:1 (or 0x3549:1 for axis B)				
Setting range (min. value | unit | max. value)			Lenze setting	
0.0	°C	255.0	25.0 °C	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	INTEGER_16

Subindex 7: **Spec. charact.: temperature sampling point 2**				
From version 01.03 The special thermal sensor characteristic is selected by setting "3" in 0x2D49:1 (or 0x3549:1 for axis B)				
Setting range (min. value | unit | max. value)			Lenze setting	
0.0	°C	255.0	150.0 °C	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	INTEGER_16

Subindex 8: **Spec. charact.: resistance sampling point 1**				
From version 01.03 The special thermal sensor characteristic is selected by setting "3" in 0x2D49:1 (or 0x3549:1 for axis B)				
Setting range (min. value | unit | max. value)			Lenze setting	
0	Ohm	30000	1000 Ohm	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				INTEGER_16

Subindex 9: **Spec. charact.: resistance sampling point 2**				
From version 01.03 The special thermal sensor characteristic is selected by setting "3" in 0x2D49:1 (or 0x3549:1 for axis B)				
Setting range (min. value | unit | max. value)			Lenze setting	
0	Ohm	30000	2225 Ohm	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				INTEGER_16

8.5.1 Spec. charact. curve for motor temperature sensor

This function extension is available from version 01.03!

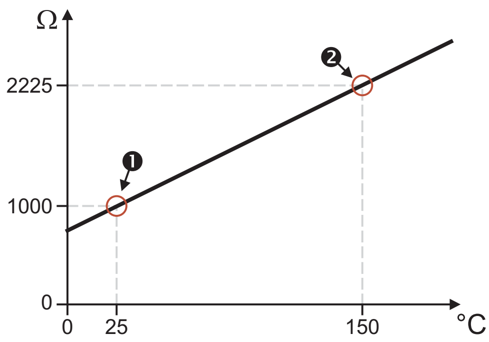
If required, you can define and activate a special characteristic for the motor temperature sensor.

Stop!

The implemented solution is not suitable as a replacement for a tripping unit for the thermal protection of rotating electrical machines (EN 60947-8:2003)!

- The special characteristic is activated by setting 0x2D49:1 = "3" (or 0x3549:1 = "3" for axis B).
- The special characteristic is defined based on two parameterisable sampling points. The two sampling points define a line which is extrapolated to the right and to the left. In the Lenze setting, the special characteristic is defined as follows:

Lenze setting of the special characteristic



Axis A	Axis B	Description	Lenze setting
0x2D49:6	0x3549:6	Sampling point ❶ - Temperature	25.0 °C
0x2D49:8	0x3549:8	Sampling point ❶ - resistance	1000 Ohm
0x2D49:7	0x3549:7	Sampling point ❷ - Temperature	150.0 °C
0x2D49:9	0x3549:9	Sampling point ❷ - resistance	2225 Ohms

Note!

- Selecting a motor from the motor catalogue causes the parameters of the special characteristic to be overwritten!
- If the inverter measures a resistance of below 122 Ω, this is assessed as a sensor error and a temperature of -666 °C is output.
- In individual cases, a short circuit is a wanted status (e.g. temperature contact, closed below 140 °C). For this purpose, the sampling point 1 (0x2D49:8 or 0x3549:8 for axis B) must be below 122 Ω to not trigger a sensor error anymore. The temperature is continued to be calculated.

8.6 Motor speed monitoring

0x2D44 | 0x3544 - Motor speed monitoring

Sub.	Name	Lenze setting	Data type
▸ 1	Motor speed monitoring: Threshold	8000 r/min	UNSIGNED_16
▸ 2	Motor speed monitoring: Response	1: Trouble	UNSIGNED_8

Subindex 1: **Motor speed monitoring: Threshold**				
Warning/error threshold for motor speed monitoring				
Setting range (min. value | unit | max. value)			Lenze setting	
50	r/min	50000	8000 r/min	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_16

Subindex 2: **Motor speed monitoring: Response**			
Response when tripping the motor speed monitoring			
Selection list (Lenze setting printed in bold)			
0	No response		
1	**Fault**		
2	Warning		
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_8

8.7 Motor phase failure monitoring

Note!

In the Lenze setting, the monitoring function is not activated!

0x2D45 | 0x3545 - Motor phase failure detection

Sub.	Name	Lenze setting	Data type
1	Motor phase failure 1: Response	0: No Response	UNSIGNED_8
2	Motor phase failure 1: Current threshold	5.0 %	UNSIGNED_8
3	Motor phase failure 1: Voltage threshold	10.0 V	UNSIGNED_16
4	Motor phase failure 2: Response	0: No Response	UNSIGNED_8

Subindex 1: **Motor phase failure 1: Response**		
Monitoring 1 (in the "operation enabled" status): Response during activation		
Selection list (Lenze setting printed in bold)		
0	**No response**	
1	Fault	
2	Warning	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX		UNSIGNED_8

Subindex 2: **Motor phase failure 1: Current threshold**				
Monitoring 1 (in the "operation enabled" status): Selection of the current threshold for activation • 100 % ≡ maximum axis current (0x2DDF:2 or 0x35DF:2 for axis B) • Background: In order to be able to reliably detect the failure of a motor phase, a certain motor current for the current sensors must flow first. The monitoring function therefore is only activated if, in the case of a servo control, the setpoint of the motor current, and in the case of a V/f characteristic control, the actual value of the motor current (display in 0x6078 or 0x6878 for axis B) has exceeded the current threshold set here.				
Setting range (min. value | unit | max. value)			Lenze setting	
1.0	%	10.0	5.0 %	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	UNSIGNED_8

Subindex 3: **Motor phase failure 1: Voltage threshold**				

Monitoring 1 (in the "operation enabled" status): Selection of the voltage threshold for monitoring several motor phases
• The V/f characteristic control enables the detection of several failed motor phases during operation.
• Monitoring with regard to a failure of several motor phases is active if
  • a response other than "0: No response" is set in subindex 1, and
  • the motor voltage exceeds the voltage threshold set here.
• The "Motor disconnected" error message is output if the motor current is lower than the device-dependent threshold value for more than 20 ms.
• The monitoring function for the failure of several motor phases can be deactivated if the value "1000.0 V" is set here.				

Setting range (min. value | unit | max. value)			Lenze setting	
0.0	V	100.0	10.0 V	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	UNSIGNED_16

Subindex 4: **Motor phase failure 2: Response**		
Monitoring 2 (in the "operation enabled" status): Response during activation		
Selection list (Lenze setting printed in bold)		
0	**No response**	
1	Fault	
2	Warning	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX		UNSIGNED_8

8.7.1 Limits of the motor phase failure monitoring

The motor phase failure monitoring can be activated for both synchronous and asynchronous motors. However, it is possible that a current flow cannot be detected for sure in the case of certain operating states of synchronous motors that are connected correctly. Hence, a fault is triggered. The following table provides an overview:

Operating status		Synchronous motor	Async. motor
Check of the motor phases prior to operation		●	●
Check of the motor phases during operation			
I_q < current threshold value	at standstill	○	●
	when motor is rotating	○	●
I_q ≥ current threshold value	at standstill	○	●
	when motor is rotating	●	●
Field weakening	when motor is rotating	●	○

● Phase failure is detected for sure

○ The monitoring function may be activated without a fault pending.

I_q Torque-forming current component

Note!

Monitoring during operation serves especially for applications which are operated with constant load and speed. In all other cases, transient processes or unfavourable operating points can cause maloperation.

Special case: Hoist

The motor phase failure monitoring may trigger a fault message in a hoist if the asynchronous motor applied for this purpose reaches the following working point:

- The hoist moves downwards, i.e. the motor is in generator mode.
- The slip frequency equals the field frequency in terms of amount. Both frequencies mutually neutralise themselves due to their opposite effective directions.

8.7.2 Monitoring 2: In the "enable operation" state transition

On the basis of test signals, this extended monitoring function for motor phase failure can detect a phase failure and check the presence of the motor. Only after a successful check, the actual operation is continued.

- Monitoring is only active for a short time after controller enable (when the device status changes from "Switched on" to "Operation enabled") if
  - a response other than "0: No response" has been set for this monitoring function in object 0x2D45:4 (or 0x3545:4 for axis B), and
  - no test mode and no identification mode (0x2825 or 0x3025 for axis B) are active.
- Before the actual operation, the motor is supplied with a maximum DC current, the max. level of which can correspond to the lower of the following two values:

$$50\,\% \cdot \sqrt{2} \cdot \text{Rated device current}$$

or

$$50\,\% \cdot \sqrt{2} \cdot \text{Rated motor current}$$

- The response set is triggered if one or several motor phase currents have not reached a specific threshold value within 5 ms after controller enable. The threshold value depends on the maximum device current and cannot be parameterised.
- The check is completed successfully if all three motor phase currents have exceeded the threshold value. Then the actual operation is continued immediately.

Note!

- As the check is cancelled immediately if all three motor phase currents have exceeded the threshold value, the setpoint current usually is not achieved.
- In order to be able to achieve the threshold value used for the check, the rated motor current must at least be 10 % of the maximum device current.
- This monitoring is independent of the further rotation of the commutation angle.

8.7.3 Monitoring 1: In "operation enabled" status

If a current-carrying motor phase (U, V, W) fails during operation, the response set for this monitoring is tripped if two conditions are met:

- Condition 1: Monitoring is activated
  - In order to be able to reliably detect the failure of a motor phase, a certain motor current for the current sensors must flow first. The monitoring function therefore is only activated if, in the case of a servo control, the setpoint of the motor current, and in the case of a V/f characteristic control, the actual value of the motor current (display in 0x6078 or 0x6878 for axis B) has exceeded a parameterisable current threshold.
- Condition 2: A specific commutation angle has been covered without the detection of a current flow.
  - In this case monitoring works according to the principle of checking for each motor phase that a current flows depending on the commutation angle.
  - Monitoring responds if a rotating field is output and hence a specific commutation angle (approx. 150°, electric) is covered without the current having exceeded a (non-parameterisable) threshold that depends on the device power.

> **Note!**
>
> The dependence on the commutation angle also causes a dependence on the motor type used:
> - The commutation angle and the angle at the shaft (number of pole pairs) of a synchronous motor are proportional. This makes it possible to predict which shaft angle is maximally covered in the event of an error.
> - There is still a slip between the commutation angle and the angle at the shaft of an asynchronous motor. This results in a load dependency which makes it impossible to predict a maximally covered shaft angle in the event of an error.
>
> For some applications (e.g. when a hoist is lowered at non-zero speed) it may happen that there is no rotating field anymore, but a DC current is flowing. In this case, condition 2 is no longer met.

8.7.4 Monitoring with regard to short circuit and earth fault

The motor phases are monitored with regard to short circuit and earth fault by means of a hardware circuit. If this monitoring function is activated, the i700 servo inverter reports an error and changes to the "Pulse inhibit" error status. This error can only be reset after an inhibit time of 5 s has elapsed.

The following table lists possible causes of a short-circuit message:

Cause	Remedies
The is a physical connection between two motor phases.	Correct the wiring and remove the short circuit.
A transient current control process causes the phase currents to increase above the signalling threshold. This can happen if it is operated at the current limit of the device and a) the current controller is set incorrectly or b) a synchronous motor is operated in the field weakening range and the current controller feedforward control in 0x2941 (or 0x3141 for axis B) has been deactivated. A transient current control process can be triggered, for instance, if an EtherCAT fault causes a quick stop. In such a situation, normally some bus cycles pass by until the error is triggered. During this period, the setpoints remain frozen which causes an unsteady setpoint profile and thus a dominant current control process.	Remedy for a): Set current controller according to alignment instructions. ▶ Setting and optimising the current controller Remedy for b): If a synchronous motor is operated at the limits or within the field weakening range, activate the current controller feedforward control in 0x2941 (or 0x3141 for axis B).

8.8 Monitoring of the ultimate motor current

Stop!

The parameterisable ultimate motor current I_{ULT} is a limit value the exceedance of which causes non-reversible damages of the motor!

Examples:
- Demagnetisation of single rotor magnets when the PM servo motor is operated.
- Destruction of the stator winding.

This limit value must not be travelled cyclically in the drive process!

The maximum current (0x6073 or 0x6873 for axis B) parameterised should provide a sufficient distance to this ultimate motor current.

Note!

When you select a Lenze motor from the catalogue and transfer the parameters of the motor to the controller, the setting of the limit value I_{ULT} is automatically adapted to the motor selected.

In case of motors of other manufacturers, we recommend the parameterisation of the limit value I_{ULT} in 0x2D46:1 (or 0x3546:1 for axis B) to be executed as one of the first commissioning steps.

If the instantaneous value of the motor current exceeds the limit value set, the response set for this monitoring function is triggered in order to protect the motor.

0x2D46 | 0x3546 - Monitoring: Ultimate motor current

Sub.	Name	Lenze setting	Data type
1	Ultimate motor current: Threshold	5.4 A	UNSIGNED_16
2	Ultimate motor current: Response	1: Trouble	UNSIGNED_8

Subindex 1: **Ultimate motor current: Threshold**				
Warning/error threshold for motor current monitoring				
Setting range (min. value | unit | max. value)			Lenze setting	
0.0	A	500.0	5.4 A	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			Scaling: 1/10	UNSIGNED_16

Subindex 2: **Ultimate motor current: Response**			
Response when tripping the motor current monitoring			
Selection list (Lenze setting printed in bold)			
0	No response		
1	**Fault**		
2	Warning		
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX			UNSIGNED_8

9 Diagnostics & error management

This chapter contains information on the drive diagnostics, error handling, and fault analysis.

Objects described in this chapter

Object Axis A	 Axis B	Name	Data type
Identification data			
0x1000		Device: Type	UNSIGNED_32
0x1001		Error memory	UNSIGNED_8
0x1008		ECAT: Manufacturer device name	STRING(50)
0x1009		Device: Hardware version	STRING(50)
0x100A		Device: Software version	STRING(50)
0x1018		ECAT: Identity object	RECORD
Error management			
0x10F3		Diagnostics: History buffer	RECORD
0x2826	0x3026	Quick stop: Duration in case of trouble	UNSIGNED_32
0x2840	0x3040	Delay time: Reset error	INTEGER_32
0x2841	0x3041	Reset error	UNSIGNED_8
0x284F	0x304F	Current fault	ARRAY [0..63] OF BYTE
0x603F	0x683F	Error code	UNSIGNED_16
0x605E	0x685E	Setting/response in the event of an error	INTEGER_16
Diagnostics parameters			
0x10F8		ECAT DC: Current time	UNSIGNED_64
0x2D81	0x3581	Counter: Operating time	RECORD
0x2D82	0x3582	Motor: Actual voltage - Veff, phase-phase	UNSIGNED_32
0x2D83	0x3583	Motor: Phase currents	RECORD
0x2D8A	0x358A	Speed monitoring: Actual speed error	INTEGER_32

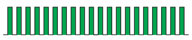
9.1 LED status displays

The LED status displays on the front of the i700 servo inverter provide quick information on some operating states.

- The two LEDs "RDY" and "ERR" serve to indicate the device status.

LED	Status	Meaning
RDY	Off	24 V supply voltage missing
		Single axis: Axis is inhibited Double axis: Both axes are inhibited
		Single axis: Axis is enabled Double axis: One or both axes are enabled
ERR	Off	No error.
		Single axis: Device error or axis error Double axis: Device error or axis error in one axis or both axes

- Three green LEDs at the EtherCAT interfaces (RJ45 sockets X4 and X5) serve to indicate the EtherCAT bus status and the connection status of the input and output socket. The arrangement of the LEDs can be seen from the "Servo-Inverter i700" hardware manual.

LED	Status	Meaning
RUN	Off	EtherCAT status "Init"
		EtherCAT state "Pre-Operational"
		EtherCAT state "Safe-Operational"
		EtherCAT state "Operational"
		EtherCAT state "Bootstrap"
L/A	Off	No EtherCAT connection
		EtherCAT communication active
		EtherCAT connection available

9.2 Indication of fault and warning (error code)

The indication of fault and warning is implemented according to CiA 301/402:

- In the error-free state, error code "0" is displayed.
- The fault/warning that has occurred first is always displayed.
- Faults overwrite warnings.
- In general, warnings do not lock up the system. If the cause of the warning disappears, the i700 servo inverter resets the warning display and bit 7 in the Statusword (0x6041 or 0x6841 for axis B) after the PDO update has been carried out in the direction of control flow.

0x603F | 0x683F - Error code

Display area (min. value | unit | max. value)			Initialisation	
0x0000		0xFFFF	0x0000	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX				UNSIGNED_16

Tip!

Chapter "CiA402 error codes / error messages" lists all possible error codes of the i700 servo inverter including causes and possible remedies.

0x284F | 0x304F - Current fault

From version 01.06
Text display of the current fault
- This object contains a reference to the text to be displayed from the ESI file as well as the replacement values for all wildcards in this text.
- The text has the same structure as the diagnostics messages in the history buffer.

☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX		ARRAY [0..63] OF BYTE

Document "ETG.1020 Protocol Enhancements"

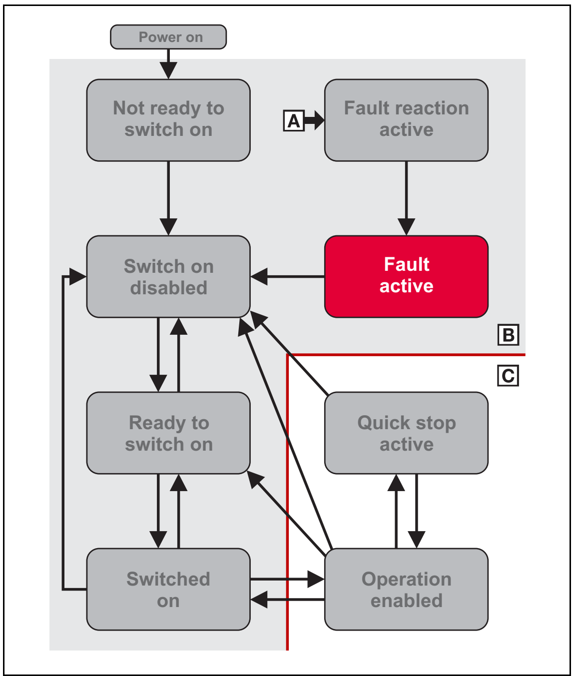
See chapter 13.3 of document "ETG.1020 Protocol Enhancements" provided by the EtherCAT Technology Group (ETG) for detailed information on the structure of the diagnostic messages.

Further indications of faults and warnings

- In the Statusword (0x6041 or 0x6841 for axis B)
  - a fault is indicated via bit 3,
  - a warning is indicated via bit 7.
- Further diagnostics can be carried out with the history buffer (0x10F3). It contains the last 32 messages of the i700 servo inverter.

9.2.1 Response of the device in the event of an error

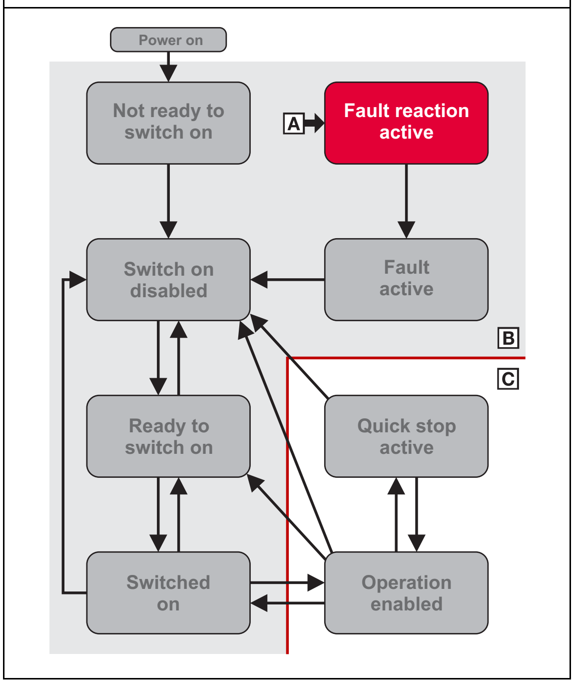
There are two different interference classes in the i700 servo inverter:



Class I: Fatal errors

Evading the "Fault reaction active" device status, errors of class I immediately cause the "Fault" device status.

The i700 servo inverter is switched off immediately (pulse inhibit) so that a guided standstill can only be realised by using a mechanical brake.



Class II: Light errors

In case of errors of class II, the error response selected in the object 0x605E (or 0x685E for axis B) takes place in the Fault reaction active.

This classification enables the i700 servo inverter to actively support the machines to reach standstill when the following light errors occur.

Light errors of class II are:

- Error code 0x4380: Motor temperature sensor
- Error code 0x7303: Error in feedback system (from version 01.06)
- Error code 0x7380: Hiperface communication error
- Error code 0x8181: EtherCAT communication
- Error code 0x8700: Sync controller

0x605E | 0x685E - Response to error

From version 01.05
Selection of the response to errors of class II:
-2 = quick stop if possible, otherwise DC current or short-circuit braking.
0 = pulse inhibit
2 = quick stop
Further information on the respective response can be found in the "Info" column further down.

Setting range (min. value | unit | max. value)			Lenze setting	
-32768		32767	-2	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				INTEGER_16

Setting/response in the event of an error		Info
-2	**Quick stop if possible, otherwise DC current or short-circuit braking**	In case of servo control, first quick stop is activated as error response. If an encoder error is detected here, ASM servo control immediately switches to the "DC-injection braking" and SM servo control immediately switches to the "short-circuit braking" function. • This setting is reasonable for drives without service brake and with limited travel way. • The braking is maintained for the entire monitoring time (0x2826 or 0x3026 for axis B) since a standstill of the motor cannot be recognised due to the missing feedback. • Afterwards, the drive changes to the "Fault" device status and inhibits the inverter. If a holding brake is parameterised and connected, it will be applied. • The braking current for DC-injection braking (ASM) has to be set in 0x2B80 (or 0x3380 for axis B).
		Background: The servo control requires a feedback of position/speed signals of the motor encoder. This basic condition also applies when the "quick stop" function is activated. If the i700 servo inverter has no access to this information anymore, it cannot realise a guided operation of the motor into the drift-free standstill. When synchronous motors are operated, the i700 servo inverter loses the current pole position in addition, which can lead to a non-guided behaviour.
2	Quick stop	Quick stop is activated as error response. • The deceleration for quick stop can be set in 0x6085 (or 0x6885 for axis B). • After the deceleration has stopped or the monitoring time for quick stop has elapsed (0x2826 or 0x3026 for axis B), the drive changes to the "Fault" device status and inhibits the inverter. If a holding brake is parameterised and connected, it will be applied.
0	Disable voltage	Evading the "Fault reaction active" device status, it is immediately changed to the "Fault" device status. • The drive is immediately switched off (pulse inhibit). • Thus the behaviour is identical to the behaviour in case of fatal errors (class I).

0x2826 | 0x3026 - Quick stop: Duration in the event of a fault

In parallel to the execution of a quick stop in the "Fault reaction active" device status, this timer is started. Depending on which of the following two conditions occurs first, the device status changes from "Fault reaction active" to "Fault":

- Condition 1: Quick stop has braked the drive to standstill.
- Condition 2: This timer has elapsed.

The status change to "Fault" triggers a pulse inhibit and, if available, the holding brake is applied.

Note!

In case of setting "-2: Quick stop if possible..." in 0x605E (or 0x685E for axis B):
If due to an encoder error, quick stop is replaced by DC current braking or short-circuit braking, a standstill cannot be detected anymore. In this case, only the condition 2 (timer has elapsed) causes an exit of the "Fault reaction active" device status.

Setting range (min. value | unit | max. value)			Lenze setting	
0	s	100	4 s	
☑ Write access ☐ CINH ☐ OSC ☑ P ☐ RX ☐ TX				UNSIGNED_32

9.2.2 Resetting the error/fault

- A fault can only be reset via bit 7 in the Controlword (0x6040 or 0x6840 for axis B).
  - If there is another fault pending after resetting the fault, bit 3 in the status word remains set and object 0x603F (or 0x683F for axis B) now shows the error code of this fault.
- Certain errors can only be reset by 24 V mains switching. These are, for instance, internal errors due to internal communication faults, initialisation errors, and checksum errors of the firmware or the persistent data (elapsed-hour meters and power-on hour meters, calibration values, etc.).
- Certain errors (e.g. earth fault or short circuit of the motor phases) can cause a delay time. In this case, the fault can only be reset via bit 7 in the Controlword after the delay time has elapsed.
  - Active delay times are displayed via bit 14 in the Lenze status word (0x2831 or 0x3031 for axis B).
  - The remaining delay time can be read out via the following object:

0x2840 | 0x3040 - Delay time: Reset error

Display area (min. value | unit | max. value)			Initialisation	
-2147483648	ms	2147483647	0 ms	
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX				INTEGER_32

0x2841 | 0x3041 - Reset error

Setting range (min. value | unit | max. value)			Lenze setting	
0		1	0	
☑ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX				UNSIGNED_8

9.3 History buffer

Via the history buffer (0x10F3), the controller can access the last 32 messages of the i700 servo inverter.

- The history buffer is saved persistently to the i700 servo inverter.
- The structure of the history buffer corresponds to a ring buffer:
  - As long as there is free history buffer space available, a message is placed in the next free slot in the buffer.
  - If all buffer slots are full, the oldest message is deleted for a new one.
  - The latest messages will always remain available.
- See the "Diag code" (32-bit word) of every single message to find out which axis the message refers to:

Diag code bits 0 ... 15	Diag code bits 16 ... 31
0xE000 ➞ device error/warning	CiA402 error code (device-specific)
0xE001 ➞ fault/warning axis A	CiA402 error code (axis-specific)
0xE002 ➞ fault/warning axis B	CiA402 error code (axis-specific)

▶ CiA402 error codes / error messages (📖 267)

Document "ETG.1020 Protocol Enhancements"

See chapter 13.3 of document "ETG.1020 Protocol Enhancements" provided by the EtherCAT Technology Group (ETG) for detailed information on the structure of the diagnostic messages.

0x10F3 - Diagnostics: History buffer

Sub.	Name	Lenze setting	Data type
▶ 1	Max. number of messages	32	UNSIGNED_8
▶ 2	Latest message	0	UNSIGNED_8
▶ 3	Latest acknowledged message	0	UNSIGNED_8
▶ 4	New active message	0	UNSIGNED_8
▶ 5	Control bits	0	UNSIGNED_16

Subindex 1: **Max. number of messages**				
Maximum number of messages which can be stored in the history buffer (from subindex 6).				
Display area (min. value | unit | max. value)			Initialisation	
0		255	32	
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX				UNSIGNED_8

Subindex 2: **Latest message**				
Subindex of the most recent message				
Display area (min. value | unit | max. value)			Initialisation	
0		255	0	
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX				UNSIGNED_8

Subindex 3: **Latest acknowledged message**				
Subindex of the most recent message acknowledged by the EtherCAT master.				
Setting range (min. value | unit | max. value)			Lenze setting	
0		255	0	
☑ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX				UNSIGNED_8

Subindex 4: **New active message**				
TRUE if messages are pending that have not been acknowledged by the EtherCAT master yet.				
Display area (min. value | unit | max. value)			Initialisation	
0		255	0	
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☑ TX				UNSIGNED_8

Subindex 5: **Control bits**				
Settings for the transmission and storage of the messages				
Setting range (min. value | unit | max. value)			Lenze setting	
0		65535	0	
☑ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX				UNSIGNED_16

9.4 CiA402 error codes / error messages

Note!

- When an error message is output, the source (device, axis A or axis B) is preceded.
- This data [value] of an error message text serves as wildcard for the value detected during the runtime of the i700 servo inverter.
- The indices for axis A and axis B are given directly in succession in the following descriptions for a better readability (index of axis A | index of axis B)

Error code 0x0000: No error

Text ID: 0x00

Cause	Remedies	Response
-	-	-

Error code 0x2320: Short circuit or earth leakage on motor side

Text ID: 0x04

Cause	Remedies	Response
• Earth fault in motor cable. • Excessive capacitive charging current in the motor cable.	• Check motor cable. • Use motor cable that is shorter or has a lower capacitance.	Fault **Note:** Error reset only possible after 5 s.

Error code 0x2340: Short circuit on motor side

Text ID: 0x05

Cause	Remedies	Response
• Short circuit/earth fault in motor cable. • Excessive capacitive charging current in the motor cable.	• Check motor cable. • Use motor cable that is shorter or has a lower capacitance.	Fault **Note:** Error reset only possible after 5 s.

Error code 0x2351: Motor utilisation (I²xt) > [value] %

Text ID: 0x38

Cause	Remedies	Response
Motor is thermally overloaded, e.g. due to: • impermissible continuous current. • Frequent or too long acceleration processes.	• Check drive dimensioning. • Check setting or the warning or error threshold.	Warning • Warning threshold can be adapted in 0x2D4E | 0x354E. From version 01.05, an error response can also be parameterised in 0x2D50:1 | 0x3550:1. • Error threshold can be adapted in 0x2D50:2 | 0x3550:2.

▶ Monitoring of the motor utilisation (I²xt)

Error code 0x2380: Fault - utilisation of the power section (Ixt) too high

Text ID: 0x06

Cause	Remedies	Response
Frequent and too long acceleration processes with overcurrent.	Check drive dimensioning.	Fault

Error code 0x2381: Warning - Utilisation of the power section (Ixt) too high

Text ID: 0x07

Cause	Remedies	Response
Frequent and too long acceleration processes with overcurrent.	Check drive dimensioning.	Warning

Error code 0x2382: Fault - device utilisation (Ixt) too high

Text ID: 0x08

Cause	Remedies	Response
Frequent and too long acceleration processes with overcurrent.	Check drive dimensioning.	Fault

Error code 0x2383: Warning - device utilisation (Ixt) too high

Text ID: 0x09

Cause	Remedies	Response
Frequent and too long acceleration processes with overcurrent.	Check drive dimensioning.	Warning

Error code 0x2384: Ultimate motor current reached

Text ID: 0x22

Cause	Remedies	Response
The motor current has exceeded the "ultimate" motor current. The "ultimate" motor current corresponds to the current level at which the permanent magnets of a synchronous motor suffer permanent damage.	Reduce maximum current (0x6073 | 0x6873) and/or adjust the current controller (0x2942 | 0x3142) in order to reduce the current overshoot.	Fault (adjustable in 0x2D46:2 | 0x3546:2) • Threshold can be adapted in 0x2D46:1 | 0x3546:1.

Error code 0x2385: Output current > [value] A (maximum device current)

Text ID: 0x24

Cause	Remedies	Response
The desired maximum current (0x6073 | 0x6873) is higher than the maximum current of the axis (display in 0x2DDF:2 | 0x35DF:2).	The permissible maximum current (0x6073 | 0x6873) is: $\frac{\text{Maximum axis current}}{\text{Rated motor current}}$ • Setting of rated motor current in 0x6075 | 0x6875.	Warning

Error code 0x2386: Clamp is active

Text ID: 0x2F

Cause	Remedies	Response
Maximum current of the axis (display in 0x2DDF:2 | 0x35DF:2) has been reached.	• Select a flatter speed ramp. • Reduce the load. • Set the Imax controller (0x2B08 | 0x3308) more dynamically.	Warning

Error code 0x2387: Clamp responded too often

Text ID: 0x30

Cause	Remedies	Response
Maximum current of the axis (display in 0x2DDF:2 | 0x35DF:2) was reached too often in a row.	• Select a flatter speed ramp. • Reduce the load. • Set the Imax controller (0x2B08 | 0x3308) more dynamically.	Fault

Error code 0x3210: DC link circuit - overvoltage [value] V

Text ID: 0x0A
Delay time: 5 ms

Cause	Remedies	Response
Due to a too high braking energy, the DC-bus voltage exceeds the overvoltage threshold which results from the mains voltage setting.	• Connect brake resistor to power supply module or use regenerative module. • Check mains setting. • Check drive profile.	Warning or fault (depending on the height of the voltage) • Warning threshold adaptable in 0x2540:5. • Fixed error threshold.

Error code 0x3220: DC link circuit - undervoltage [value] V

Text ID: 0x0B

Cause	Remedies	Response
DC bus voltage is lower than the undervoltage threshold resulting from the mains setting.	• Check mains voltage. • Check DC-bus voltage. • Check mains settings.	Warning or fault (depending on the height of the voltage) • Warning threshold adaptable in 0x2540:2. • Fixed error threshold.

Error code 0x4210: Module temperature too high, [value]

Text ID: 0x0C

Cause	Remedies	Response
Heatsink temperature higher than fixed temperature limit (100 °C): • Ambient controller temperature too high. • Dirty fan or ventilation slots. • Fan defective. • Warning threshold set too low.	• Check control cabinet temperature. • Clean filter. • Clean controller. • Replace fan.	Warning or fault (depending on the height of the temperature) • Warning threshold can be adapted in 0x2D84:2 | 0x3584:2. • Fixed error threshold.

▶ Monitoring of the heatsink temperature

Error code 0x4280: Internal fault - module temperature monitoring

Text ID: 0x26

Cause	Remedies	Response
Thermal sensor error	• 24 V mains switching required. • If the error occurs frequently, please contact Lenze.	Fault

Error code 0x4310: Motor temperature too high, [value]

Text ID: 0x0D

Cause	Remedies	Response
Motor temperature higher than variable temperature limit: • Motor too hot due to impermissibly high currents or frequent and too long acceleration processes. • No KTY connected. • Error or warning threshold set too low.	• Check drive dimensioning. • Connect KTY or switch off monitoring. • Set a higher warning or error threshold.	Fault (adjustable in 0x2D49:2 | 0x3549:2) • Warning threshold can be adapted in 0x2D49:3 | 0x3549:3. • Error threshold adaptable in 0x2D49:4 | 0x3549:4.

▸ Motor temperature monitoring

Error code 0x4380: Motor temperature sensor

Text ID: 0x25

Cause	Remedies	Response
Motor temperature sensor	• Connect KTY or switch off monitoring. • Set a higher warning or error threshold.	Class II fault (adjustable in 0x2D49:2 | 0x3549:2) • If "fault" is set, the response set in 0x605E | 0x685E takes place. • Warning threshold can be adapted in 0x2D49:3 | 0x3549:3. • Error threshold adaptable in 0x2D49:4 | 0x3549:4.

▸ Motor temperature monitoring

Error code 0x5112: 24-V supply

Text ID: 0x0E

Cause	Remedies	Response
24 V supply has failed or has fallen below the warning threshold (21.45 V) or error threshold (18.15 V).	Check 24 V voltage supply.	Warning or fault (depending on the height of the voltage) • Thresholds are fixed.

Error code 0x6010: Watchdog reset

Text ID: 0x0F

Cause	Remedies	Response
• Reset was triggered via device command 1100. • Internal error	If the reset was not triggered via the device command, consult Lenze if the error occurs frequently.	Warning if restart of the drive is requested via command. Otherwise fault.

Error code 0x6310: Incorrect parameter set download

Text ID: 0x10

Cause	Remedies	Response
Error when downloading a parameter set: • CRC error in the parameter set detected. • Vendor, product code or revision of the object directory are different. • Unknown parameter set.	• Repeat parameter set download. • Recreate parameter set.	Warning if single parameters cannot be imported. Fault if all parameters cannot be imported.

Error code 0x6320: Parameter error in object [value]

Text ID: 0x11

Cause	Remedies	Response
Unknown index in the parameter set.	Recreate parameter set.	Warning

Error code 0x7121: PLI - motor blocked

Text ID: 0x12

Cause	Remedies	Response
An error occurred during the pole position identification. The pole position identification could not be completed successfully: • Too many deviations during the identification. • Motor is blocked.	• Check whether all requirements for an identification of the pole position are fulfilled. • Ensure that the machine is not braked or blocked during the pole position identification. • Repeat the pole position identification.	Fault (adjustable in 0x2C60 | 0x3460)

▶ Synchronous motor (SM): Pole position identification

Error code 0x7303: Error in feedback system

Text ID: 0x13

Cause	Remedies	Response
• Encoder cable interrupted. • Encoder defective.	• Check encoder cable. • Check encoder. • Switch off monitoring if feedback is not used.	Warning in the states "Not ready to switch on" and "Switch on disabled". Otherwise class II fault (adjustable in 0x2C45 | 0x3445) • If "fault" is set, the response set in 0x605E | 0x685E takes place.

▶ Encoder open-circuit monitoring

Error code 0x7380: Hiperface communication error

Text ID: 0x3A

Cause	Remedies	Response
Communication with HIPERFACE® absolute value encoder is disturbed.	• Check the supply voltage of the encoder (0x2C42:2 | 0x3442:2). • Check wiring to the encoder.	Up to and including version 01.05: Fault (adjustable in 0x2C41:4 | 0x3441:4) From version 01.06: Class II fault (adjustable in 0x2C41:4 | 0x3441:4) • If "fault" is set, the response set in 0x605E | 0x685E takes place.

Error code 0x7381: Hiperface: Invalid absolute value - motor speed too high

Text ID: 0x41

Cause	Remedies	Response
The absolute position of the HIPERFACE® absolute value encoder cannot be accepted as the motor rotates too fast.	Let the motor coast.	Fault (adjustable in 0x2C41:4 | 0x3441:4)

Error code 0x8180: ECAT DC - synchronisation required

Text ID: 0x15

Cause	Remedies	Response
EtherCAT DC mode not activated. (DC = Distributed Clock)	Activate DC mode.	Warning

Error code 0x8181: EtherCAT communication

Text ID: 0x19

Cause	Remedies	Response
EtherCAT communication error	• Check EtherCAT bus cabling. • Select a longer PDI watchdog time.	Class II fault • The response set in 0x605E | 0x685E takes place.

Error code 0x8280: Sync Manager - address [value]

Text ID: 0x16

Cause	Remedies	Response
Incorrect Sync Manager setting of the EtherCAT master.	• Check the devices on the bus. • Check cabling or sequence of the devices on the bus. • Remove the i700 servo inverter from the project tree and rescan the bus. • Check the device description of the i700 servo inverter.	Fault

Error code 0x8281: Sync Manager - actual size [value]

Text ID: 0x17

Cause	Remedies	Response
Incorrect Sync Manager setting of the EtherCAT master.	• Check the devices on the bus. • Check cabling or sequence of the devices on the bus. • Remove the i700 servo inverter from the project tree and rescan the bus. • Check the device description of the i700 servo inverter.	Fault

Error code 0x8282: Sync Manager - settings [value]

Text ID: 0x18

Cause	Remedies	Response
Incorrect Sync Manager setting of the EtherCAT master.	• Check the devices on the bus. • Check cabling or sequence of the devices on the bus. • Remove the i700 servo inverter from the project tree and rescan the bus. • Check the device description of the i700 servo inverter.	Fault

Error code 0x8283: PDO Mapping: Object unknown (index [value])

Text ID: 0x2B

Cause	Remedies	Response
Unknown index in the PDO mapping.	Check PDO mapping.	Fault

Error code 0x8284: No PDO mapping object (index [value])

Text ID: 0x2C

Cause	Remedies	Response
Index cannot be mapped as PDO.	Check PDO mapping.	Fault

Error code 0x8285: Too many mapped objects (max. number = [value])

Text ID: 0x2D

Cause	Remedies	Response
Max. number of PDO mapping objects has been exceeded (> 64).	Reduce the number of mapping objects in the "free mapping".	Fault

Error code 0x8286: PDO mapping - error

Text ID: 0x2E

Cause	Remedies	Response
Incorrect PDO mapping.	Check PDO mapping.	Fault

Error code 0x8700: Sync controller

Text ID: 0x14

Cause	Remedies	Response
• EtherCAT Sync0 signal has failed. • PDO transmitted or received too late.	• Check in EtherCAT master configuration whether DC mode is set. • Restart EtherCAT configuration. • Select a later instant of reception, outside the user shift time. • Sync Error Counter - increase monitoring limit (0x10F1:2).	Class II fault • The response set in 0x605E | 0x685E takes place.

Error code 0x8701: Time-out during synchronisation with EtherCAT Sync signal

Text ID: 0x39

Cause	Remedies	Response
During the EtherCAT state change from "Pre-Operational" to "Safe-Operational": • EtherCAT Sync0 signal has failed. • Jitter of the EtherCAT Sync0 signal is too high.	• Check EtherCAT master configuration for DC deviation. • Restart EtherCAT configuration.	Fault

Error code 0xFF00: Fatal internal error

Text ID: 0x1A

Cause	Remedies	Response
Internal error	• 24 V mains switching required. • If the error occurs frequently, please contact Lenze.	Fault

Error code 0xFF01: Fatal internal communication error, cycle [value]

Text ID: 0x1B

Cause	Remedies	Response
Internal error	• 24 V mains switching required. • If the error occurs frequently, please contact Lenze.	Fault

Error code 0xFF02: Brake

Text ID: 0x1C

Cause	Remedies	Response
Brake error due to short circuit or cable break.	Check brake cable or brake.	Warning in "manual control" mode. Otherwise class II fault (adjustable in 0x2820:6 | 0x3020:6) • If "fault" is set, the response set in 0x605E | 0x685E takes place.

Error code 0xFF03: Fatal internal error - overflow task [value]

Text ID: 0x1D

Cause	Remedies	Response
Internal error	• 24 V mains switching required. • If the error occurs frequently, please contact Lenze.	Fault

Error code 0xFF04: PLI - motor movement too large

Text ID: 0x1E

Cause	Remedies	Response
Motor moved too much during pole position identification.	• Check settings of the pole position identification in 0x2C62 | 0x3462. • If this error takes place in pole position identification without any motion, the motor has to be stalled.	Fault (adjustable in 0x2C60 | 0x3460)

▸ Synchronous motor (SM): Pole position identification

Error code 0xFF05: STO inhibited

Text ID: 0x1F

Cause	Remedies	Response
The controller is inhibited via STO terminals, despite controller enable via the control word.	• Cancel STO or inhibit controller. • Check the cabling of the STO terminals.	Warning with change to the "Switch on disabled" device status.

Error code 0xFF06: Max. speed reached

Text ID: 0x20

Cause	Remedies	Response
Motor speed too high.	• Reduce motor speed. • Check resolver or cabling.	Fault (adjustable in 0x2D44:2 | 0x3544:2) • Threshold can be adapted in 0x2D44:1 | 0x3544:1.

Error code 0xFF07: Impermissible during identification or in the test mode

Text ID: 0x21

Cause	Remedies	Response
A parameter was attempted to change which influences the identification in progress or the currently activated test mode.	Postpone changing the parameter until the identification is completed or the test mode is deactivated again.	Warning

Error code 0xFF08: Impossible during identification

Text ID: 0x23

Cause	Remedies	Response
A parameter was attempted to change which influences the identification in progress.	Postpone changing the parameter until the identification is completed.	Warning

Error code 0xFF09: Motor phase failure

Text ID: 0x27

Cause	Remedies	Response
Multiple motor phases are not connected.	• Correct the wiring between the controller and the motor. • In the case of a false alarm, increase the threshold values for monitoring in 0x2D45 | 0x3545.	Default setting: No response Monitoring takes place before and during operation. Both monitoring modes can trigger an error. For parameter setting of the two monitoring modes, see 0x2D45 | 0x3545.

▶ Motor phase failure monitoring

Error code 0xFF0A: Motor phase failure - phase U

Text ID: 0x28

Cause	Remedies	Response
Motor phase U is not connected.	• Correct the wiring between the controller and the motor. • In the case of a false alarm, increase the threshold values for monitoring in 0x2D45 | 0x3545.	Default setting: No response Monitoring takes place before and during operation. Both monitoring modes can trigger an error. For parameter setting of the two monitoring modes, see 0x2D45 | 0x3545.

▶ Motor phase failure monitoring

Error code 0xFF0B: Motor phase failure - phase V

Text ID: 0x29

Cause	Remedies	Response
Motor phase V is not connected.	• Correct the wiring between the controller and the motor. • In the case of a false alarm, increase the threshold values for monitoring in 0x2D45 | 0x3545.	Default setting: No response Monitoring takes place before and during operation. Both monitoring modes can trigger an error. For parameter setting of the two monitoring modes, see 0x2D45 | 0x3545.

▶ Motor phase failure monitoring

Error code 0xFF0C: Motor phase failure - phase W

Text ID: 0x2A

Cause	Remedies	Response
Motor phase W is not connected.	• Correct the wiring between the controller and the motor. • In the case of a false alarm, increase the threshold values for monitoring in 0x2D45 | 0x3545.	Default setting: No response Monitoring takes place before and during operation. Both monitoring modes can trigger an error. For parameter setting of the two monitoring modes, see 0x2D45 | 0x3545.

▶ Motor phase failure monitoring

Error code 0xFF0D: No resolver connected --> command cannot be executed.

Text ID: 0x31

Cause	Remedies	Response
The identification of a resolver error was attempted during V/f characteristic control without feedback system.	The resolver error can only be identified if the resolver is used as feedback system during servo control.	Warning

Error code 0xFF0E: Speed too low --> command cannot be executed.

Text ID: 0x32

Cause	Remedies	Response
The actual speed is too low for the resolver error to be identified.	Repeat identification at higher speed.	Warning

Error code 0xFF0F: No resolver connected --> command cannot be executed.

Text ID: 0x33

Cause	Remedies	Response
The identification of a resolver error was attempted but an encoder is used as feedback system.	Identification is only possible with devices with resolver feedback.	Warning

Error code 0xFF10: Time-out

Text ID: 0x34

Cause	Remedies	Response
The actual speed is too low for the resolver error to be identified.	Repeat identification at higher speed.	Warning

Error code 0xFF11: Numerical problem

Text ID: 0x35

Cause	Remedies	Response
The actual speed is too low for the resolver error to be identified.	Repeat identification at higher speed.	Warning

Error code 0xFF12: Inverter error too big

Text ID: 0x36

Cause	Remedies	Response
The results of the inverter error identification are not plausible.	Check shielding of the motor cable. Connect the shield with a surface as large as possible.	Warning

Error code 0xFF13: Identification aborted

Text ID: 0x37

Cause	Remedies	Response
A started identification was aborted due to an error or a controller inhibit.	Repeat the identification and avoid the cause of the abort.	Faulty pole position identification (adjustable in 0x2C60 | 0x3460) Otherwise warning

Error code 0xFF14: Not allowed in state 'operation enabled' or 'quick stop active'

Text ID: 0x3B

Cause	Remedies	Response
Reading out and accepting the hiperface data is only permitted when the controller is inhibited.	Inhibit controller.	Warning

Error code 0xFF15: No Hiperface absolute value encoder connected --> command cannot be executed

Text ID: 0x3C

Cause	Remedies	Response
When the "resolver" device version or the sin/cos encoder has been selected, an attempt is being made to read the data from a HIPERFACE® absolute value encoder.	The command is only possible with the "encoder" device version. Moreover, the "2: Hiperface absolute value encoder" selection has to be set in 0x2C40 | 0x3440.	Warning

Error code 0xFF16: Connected Hiperface absolute value encoder is not supported

Text ID: 0x3D

Cause	Remedies	Response
The HIPERFACE® absolute value encoder has a type code which is neither stored in the firmware nor corresponds to the type code set in 0x2C41:2 | 0x3441:2.	Set type code in 0x2C41:2 | 0x3441:2.	Fault (adjustable in 0x2C41:4 | 0x3441:4)

▸ Additional settings for SinCos absolute value encoders with HIPERFACE® protocol

Error code 0xFF17: Connected Hiperface absolute value encoder is not supported --> command cannot be executed

Text ID: 0x3E

Cause	Remedies	Response
The HIPERFACE® absolute value encoder is not supported by the device. Hence, no data can be detected.	Connect a HIPERFACE® absolute value encoder which is supported by the device.	Warning

▸ Supported encoder types with HIPERFACE® protocol

Error code 0xFF18: Communication timeout in the manual control mode

Text ID: 0x3F

Cause	Remedies	Response
In the "manual control" mode, writing the setpoint data failed to appear for a longer period of time than set in 0x2836:5 | 0x3036:5.	• Check connection between device, Controller and »EASY Starter« or »PLC Designer«. • Increase the time period in 0x2836:5 | 0x3036:5. • Restart manual jog dialog.	Fault

▸ Manual control

Error code 0xFF19: Internal error during identification

Text ID: 0x40

Cause	Remedies	Response
Internal error during identification.	Please contact Lenze.	Warning

9.5 Identification data

0x1000 - Device: Type

Selection list (read only)			
0x01020192	Single Servo Axis CiA402		
0x02020192	**Double Servo Axis CiA402**		
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			UNSIGNED_32

0x1001 - Error memory

The error memory displays the error cause in a bit coded manner. It is also possible that several errors are pending at the same time.

Display area (min. value | unit | max. value)			Initialisation	
0		255	0	
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☑ TX				UNSIGNED_8

0x1008 - ECAT: Manufacturer's device name

☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX		STRING(50)

0x1009 - Device: Hardware version

☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX		STRING(50)

0x100A - Device: Software version

☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX		STRING(50)

0x1018 - ECAT: Identification data

Note:
The fixed identification data depend on the device version of the i700 servo inverter.
Examples of identification data for a double axis are provided below.

Sub.	Name	Lenze setting	Data type
1	ECAT: Vendor ID	59	UNSIGNED_32
2	ECAT: Product code	0x69070002: i700 (Double Inverter, Safety STO)	UNSIGNED_32
3	ECAT: ESI revision	65542	UNSIGNED_32
4	ECAT: Serial number		UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

9.6 Diagnostics parameters

0x10F8 - ECAT DC: Current time

Display of the time information the i700 servo inverter is currently using (time of the device if you will).

▸ Real-time information (Distributed Clock)

Display area (min. value | unit | max. value)			Initialisation	
0	ns	2^{64}-1		
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☑ TX				UNSIGNED_64

0x2D81 | 0x3581 - Counter: Operating time

Sub.	Name	Lenze setting	Data type
▸ 1	Device: Operating time		UNSIGNED_32
▸ 2	Device: Power-on time		UNSIGNED_32

Subindex 1: **Device: Operating time**				
Display of the seconds the i700 servo inverter has been operated (device status "Operation enabled") so far.				
Display area (min. value | unit | max. value)			Initialisation	
0	s	4294967295		
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX				UNSIGNED_32

Subindex 2: **Device: Power-on time**				
Display of the seconds the i700 servo inverter has been switched on so far.				
Display area (min. value | unit | max. value)			Initialisation	
0	s	4294967295		
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX				UNSIGNED_32

0x2D82 | 0x3582 - motor: Actual voltage - Veff, phase-phase

Display of the current motor voltage

Display area (min. value | unit | max. value)			Initialisation	
0.0	V	429496729.5	0.0 V	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/10	UNSIGNED_32

0x2D83 | 0x3583 - Motor: Phase currents

Display of the current motor current of each individual motor phase

Sub.	Name	Lenze setting	Data type
1	Zero system current		INTEGER_32
2	Current - phase U		INTEGER_32
3	Current - phase V		INTEGER_32
4	Current - phase W		INTEGER_32

Subindex 1: **Zero system current**				
Display area (min. value | unit | max. value)			Initialisation	
-21474836.48	A	21474836.47		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/100	INTEGER_32

Subindex 2: **Current - phase U**				
Display area (min. value | unit | max. value)			Initialisation	
-21474836.48	A	21474836.47		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/100	INTEGER_32

Subindex 3: **Current - phase V**				
Display area (min. value | unit | max. value)			Initialisation	
-21474836.48	A	21474836.47		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/100	INTEGER_32

Subindex 4: **Current - phase W**				
Display area (min. value | unit | max. value)			Initialisation	
-21474836.48	A	21474836.47		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/100	INTEGER_32

0x2D8A | 0x358A - Speed monitoring: Actual speed error

Display area (min. value | unit | max. value)			Initialisation	
-480000	r/min	480000		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX				INTEGER_32

10 Service/internal

Objects described in this chapter

Object		Name	Data type
Axis A	Axis B		
0x2DD0	0x35D0	Field: Values	RECORD
0x2DD1	0x35D1	Motor: Currents	RECORD
0x2DD2	0x35D2	Position: Target position interpolated	INTEGER_32
0x2DD3	0x35D3	Target speeds	RECORD
0x2DD4	0x35D4	Speed controller: Output signal	RECORD
0x2DD5	0x35D5	Target torque	INTEGER_32
0x2DD6	0x35D6	Torque: Filter cascade	RECORD
0x2DD7	0x35D7	Voltage values	RECORD
0x2DDC	0x35DC	Slip: Actual slip	INTEGER_16
0x2DDD	0x35DD	Device: Actual output frequency	INTEGER_16
0x2DDE	0x35DE	Motor: Actual position of rotor angle	INTEGER_16
0x2DDF	0x35DF	Axis: Device data	RECORD

0x2DD0 | 0x35D0 - field: Values

Sub.	Name	Lenze setting	Data type
1	Field: Actual field		UNSIGNED_16
2	Field: Field setpoint	0 %	UNSIGNED_16

Subindex 1: **Field: actual field**				
Display area (min. value | unit | max. value)			Initialisation	
0	%	65535		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX				UNSIGNED_16

Subindex 2: **Field: target field**				
Display area (min. value | unit | max. value)			Initialisation	
0	%	65535	0 %	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX				UNSIGNED_16

0x2DD1 | 0x35D1 - motor: Currents

Sub.	Name	Lenze setting	Data type
1	D-current (id): Actual D-current		INTEGER_16
2	Q-current (iq): Actual Q-current		INTEGER_16
3	D-current (id): D-current setpoint		INTEGER_16
4	Q-current (iq): Q-current setpoint		INTEGER_16
5	Motor current Ieff		INTEGER_16

Subindex 1: **D current (id): Actual D-current**				
Display area (min. value | unit | max. value)			Initialisation	
-327.67	A	327.67		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/100	INTEGER_16

Subindex 2: **Q current (iq): Actual Q-current**				
Display area (min. value | unit | max. value)			Initialisation	
-327.67	A	327.67		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/100	INTEGER_16

Subindex 3: **D current (id): Target D current**				
Display area (min. value | unit | max. value)			Initialisation	
-327.67	A	327.67		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/100	INTEGER_16

Subindex 4: **Q current (iq): Target Q current**				
Display area (min. value | unit | max. value)			Initialisation	
-327.67	A	327.67		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/100	INTEGER_16

Subindex 5: **Motor current Ieff**				
Note! The r.m.s. value of the current is calculated from the time values by means of the use of the factor "√2". This also happens in case of the field frequency 0 Hz (DC current). In case of the DC injection braking, the r.m.s. value is thus given in this object by the factor "√2" too low.				
Display area (min. value | unit | max. value)			Initialisation	
-327.67	A	327.67		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/100	INTEGER_16

0x2DD2 | 0x35D2 - Position: Target position interpolated

Display area (min. value | unit | max. value)			Initialisation	
-2147483647	[Pos unit]	2147483647	0 [Pos unit]	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX				INTEGER_32

0x2DD3 | 0x35D3 - Target speeds

Sub.	Name	Lenze setting	Data type
1	Speed: Target speed	0 rpm	INTEGER_32
2	Speed: Target speed 2	0 rpm	INTEGER_32
3	Speed: Target speed limited	0 rpm	INTEGER_32

Subindex 1: **Speed: target speed**				
Display area (min. value | unit | max. value)			Initialisation	
-480000	r/min	480000	0 rpm	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX				INTEGER_32

Subindex 2: **Speed: target speed 2**				
Display area (min. value | unit | max. value)			Initialisation	
-480000	r/min	480000	0 rpm	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX				INTEGER_32

Subindex 3: **Speed: target speed limited**				
Display area (min. value | unit | max. value)			Initialisation	
-480000	r/min	480000	0 rpm	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX				INTEGER_32

0x2DD4 | 0x35D4 - Speed controller: Output signal

Sub.	Name	Lenze setting	Data type
1	Speed controller: Output signal 1	0.0 %	INTEGER_16
2	Speed controller: Output signal 2	0.0 %	INTEGER_16

Subindex 1: **speed controller: output signal 1**				
Display area (min. value | unit | max. value)			Initialisation	
-3276.7	%	3276.7	0.0 %	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/10	INTEGER_16

Subindex 2: **speed controller: output signal 2**				
Display area (min. value | unit | max. value)			Initialisation	
-3276.7	%	3276.7	0.0 %	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/10	INTEGER_16

0x2DD5 | 0x35D5 - torque: Target torque

Display area (min. value | unit | max. value)			Initialisation	
-21474836.47	Nm	21474836.47	0.00 Nm	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/100	INTEGER_32

0x2DD6 | 0x35D6 - Torque: Filter cascade

Sub.	Name	Lenze setting	Data type
1	Torque: Filter cascade - starting value	0.0 %	INTEGER_16
2	Torque: Notch filter 1 - input value	0.0 %	INTEGER_16
3	Torque: Notch filter 2 - input value	0.0 %	INTEGER_16
4	Torque: Target torque filtered	0.0 %	INTEGER_16

Subindex 1: **Torque: filter cascade - starting value**				
Display area (min. value | unit | max. value)			Initialisation	
-3276.7	%	3276.7	0.0 %	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/10	INTEGER_16

Subindex 2: **Torque: notch filter 1 - input value**				
Display area (min. value | unit | max. value)			Initialisation	
-3276.7	%	3276.7	0.0 %	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/10	INTEGER_16

Subindex 3: **Torque: notch filter 2 - input value**				
Display area (min. value | unit | max. value)			Initialisation	
-3276.7	%	3276.7	0.0 %	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/10	INTEGER_16

Subindex 4: **Torque: target torque filtered**				
Display area (min. value | unit | max. value)			Initialisation	
-3276.7	%	3276.7	0.0 %	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/10	INTEGER_16

0x2DD7 | 0x35D7 - Voltage values

Sub.	Name	Lenze setting	Data type
1	Current motor voltage limit: Actual voltage		INTEGER_16
2	D-current controller: Output signal		INTEGER_16
3	Q-current controller: Output signal		INTEGER_16
4	D-voltage (magnetisation)		INTEGER_16
5	Q-voltage (torque)		INTEGER_16

Subindex 1: **Current motor voltage limit: actual voltage**				
Display area (min. value | unit | max. value)			Initialisation	
-3276.7	V	3276.7		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/10	INTEGER_16

Subindex 2: **D-current controller: output signal**				
Display area (min. value | unit | max. value)			Initialisation	
-3276.7	V	3276.7		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/10	INTEGER_16

Subindex 3: **Q-current controller: output signal**				
Display area (min. value | unit | max. value)			Initialisation	
-3276.7	V	3276.7		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/10	INTEGER_16

Subindex 4: **D-voltage (magnetisation)**				
Display area (min. value | unit | max. value)			Initialisation	
-3276.7	V	3276.7		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/10	INTEGER_16

Subindex 5: **Q-voltage (torque)**				
Display area (min. value | unit | max. value)			Initialisation	
-3276.7	V	3276.7		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/10	INTEGER_16

0x2DDC | 0x35DC - Slip: Actual slip

Display area (min. value | unit | max. value)			Initialisation	
-3276.7	Hz	3276.7		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/10	INTEGER_16

0x2DDD | 0x35DD - Device: Actual output frequency

Display area (min. value | unit | max. value)			Initialisation	
-3276.7	Hz	3276.7	0.0 Hz	
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/10	INTEGER_16

0x2DDE | 0x35DE - motor: Actual position of rotor angle

Display area (min. value | unit | max. value)			Initialisation	
0		2047		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX				INTEGER_16

0x2DDF | 0x35DF - Axis: Device data

Sub.	Name	Lenze setting	Data type
▸ 1	Axis: Rated current		UNSIGNED_16
▸ 2	Axis: Maximum current		UNSIGNED_16
▸ 5	Axis: Supported feedback type	0: Product-defined	UNSIGNED_8

Subindex 1: **axis: rated current**				
Display area (min. value | unit | max. value)			Initialisation	
0.00	A	655.35		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/100	UNSIGNED_16

Subindex 2: **axis: maximum current**				
Display area (min. value | unit | max. value)			Initialisation	
0.00	A	655.35		
☐ Write access ☐ CINH ☑ OSC ☐ P ☐ RX ☑ TX			Scaling: 1/100	UNSIGNED_16

Subindex 5: **Axis: Supported feedback type**			
From version 01.03			
Selection list (read only)			
0	**Product-defined**		
1	No feedback system		
2	Resolver		
3	SinCos + Hiperface Encoder feedback		
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☑ TX			UNSIGNED_8

10.1 Firmware update

The drive component firmware can be replaced by means of the available EtherCAT interface (RJ45 socket) via FoE. In order to carry out a firmware download, the drive component must be in the "Bootstrap" EtherCAT status. Only one firmware with the name "i700.bin" is accepted.

11 Appendix

11.1 Table of attributes

The table of attributes contains information required for a communication with the controller via objects.

How to read the table of attributes:

Column	Entry	Meaning
Index	*0xXXXX*	Object index
Name	*Text*	Object/parameter name
	Subindex: text	Number and parameter name of the subindex
Data type	Data type of the parameter:	
	INTEGER_8	1 byte, with sign
	INTEGER_16	2 bytes with sign
	INTEGER_32	4 bytes with sign
	UNSIGNED_8	1 byte without sign
	UNSIGNED_16	2 bytes without sign
	UNSIGNED_32	4 bytes without sign
	UNSIGNED_64	8 bytes without sign
	STRING(xx)	ASCII string (with character length xx)
	ARRAY [] OF...	ARRAY
Scaling	Scaling of the parameter	
Attributes	P	Parameter can be persisted.
	OSC	Parameter can be recorded by means of the oscilloscope function.
	Tx	Parameter can be mapped into the TPDO.
	Rx	Parameter can be mapped into the RPDO.
	CINH	Parameter can only be written to if controller inhibit is set.

Index	Name	Data type	Scaling	Attributes				
Communication Area								
0x1000	Device: Type	UNSIGNED_32	-					
0x1001	Error memory	UNSIGNED_8	-				Tx	
0x1008	ECAT: Manufacturer device name	STRING(50)	-					
0x1009	Device: Hardware version	STRING(50)	-					
0x100A	Device: Software version	STRING(50)	-					
0x1018	ECAT: Identity object							
	1: ECAT: Manufacturer ID	UNSIGNED_32	-					
	2: ECAT: Product code	UNSIGNED_32	-					
	3: ECAT: ESI revision	UNSIGNED_32	-					
	4: ECAT: Serial number	UNSIGNED_32	-					
0x10F1	ECAT: Behaviour in case of error							
	1: Internal device response	UNSIGNED_32	-					
	2: Synchronisation: Error threshold	UNSIGNED_32	-	P				
Greyed out = read only								

Index	Name	Data type	Scaling	Attributes				
0x10F3	Diagnostics: History buffer							
	1: Max. number of messages	UNSIGNED_8	-					
	2: Latest message	UNSIGNED_8	-					
	3: Latest acknowledged message	UNSIGNED_8	-					
	4: Latest active message	UNSIGNED_8	-				Tx	
	5: Control bits	UNSIGNED_16	-					
0x10F8	ECAT DC: Current time	UNSIGNED_64	-				Tx	
Manufacturer specific area								
0x2000	Device: Data							
	1: Device: Product designation	STRING(50)	-					
	2: Device: Serial number	STRING(50)	-					
	3: Device: Manufacturing date	STRING(50)	-					
0x2001	Device: Name	STRING(128)	-	P				
0x2020	EoE information							
	1: Virtual MAC address	STRING(32)	-					
	2: IP address	STRING(32)	-					
	3: Subnet mask	STRING(32)	-					
	4: Default gateway	STRING(32)	-					
	5: DNS server	STRING(32)	-					
	6: DNS name	STRING(50)	-					
	7: Rx packets	UNSIGNED_32	-					
	8: Tx packets	UNSIGNED_32	-					
0x2021	Device: Optical recognition							
	1: Start optical recognition	UNSIGNED_8	-					
	2: Optical recognition: Blinking time	UNSIGNED_16	-					
0x2022	Device command	UNSIGNED_32	-					
0x2030	Parameter set: Validity check (CRC)	UNSIGNED_32	-					
0x2500	Touch probe (TP): Debounce time	UNSIGNED_16	-	P				
0x2540	Device: Voltage values							
	1: Mains: Rated voltage	UNSIGNED_8	-	P				
	2: Undervoltage (LU): Warning threshold	UNSIGNED_16	-	P				
	3: Undervoltage (LU): Error threshold	UNSIGNED_16	-	P				
	4: Undervoltage (LU): Threshold 'Reset error'	UNSIGNED_16	-	P				
	5: Overvoltage (OU): Warning threshold	UNSIGNED_16	-	P				
	6: Overvoltage (OU): Error threshold	UNSIGNED_16	-	P				
	7: Overvoltage (OU): Threshold 'Reset error'	UNSIGNED_16	-	P				
0x2580	ECAT DC: Real-time information							
	1: ECAT DC: Status real-time information	UNSIGNED_8	-					
	2: ECAT DC: Time stamp first real-time information	UNSIGNED_64	-					
	3: ECAT DC: Time stamp last real-time information	UNSIGNED_64	-					
	4: ECAT DC: Current time	UNSIGNED_64	-					
Axis A - holding brake control								
0x2820	Brake control: Settings							
	1: Brake: Mode	UNSIGNED_8	-	P		Rx		
	2: Brake: Application time	UNSIGNED_16	-	P				
	3: Brake: Release time	UNSIGNED_16	-	P				
	4: Brake: Brake recognition	UNSIGNED_16	-					
	5: Brake: Control logic	UNSIGNED_8	-	P				
	6: Brake monitoring: Response	UNSIGNED_8	-	P				
Greyed out = read only								

Index	Name	Data type	Scaling	Attributes				
Axis A - device control								
0x2822	Axis command	UNSIGNED_32	-					
0x2823	Axis command: Progress	UNSIGNED_8	-					
0x2824	Device control via PDO: Activation	UNSIGNED_8	-	P				CINH
0x2825	Mode: Selection	UNSIGNED_8	-	P				CINH
0x2826	Quick stop: Duration in case of trouble	UNSIGNED_32	-	P				
0x2830	Lenze control word	UNSIGNED_16	-		OSC	Rx		
0x2831	Lenze status word	UNSIGNED_16	-		OSC		Tx	
0x2832	Identification: Status word	UNSIGNED_16	-		OSC			
0x2833	Lenze statusword 2	UNSIGNED_16	-		OSC		Tx	
0x2835	Manual test mode: Settings							
	1: Manual test mode: Setpoint current	INTEGER_16	-		OSC	Rx		
	2: Manual test mode: Frequency	INTEGER_16	1/10		OSC	Rx		
	3: Manual test mode: Starting angle	INTEGER_16	1/10					
0x2836	Manual jog: Settings							
	1: Manual jog: Setpoint current	UNSIGNED_16	-			Rx		
	2: Manual jog: Frequency	INTEGER_16	1/10		OSC	Rx		
	3: Manual jog: Ramp time - current	UNSIGNED_16	-	P				
	4: Manual jog: Ramp time - frequency	UNSIGNED_16	-	P				
	5: Manual jog: Time monitoring	UNSIGNED_32	-	P				
	6: Manual jog: Current controller gain	UNSIGNED_32	1/100	P				
	7: Manual jog: Current controller reset time	UNSIGNED_32	1/100	P				
Axis A - error management								
0x2840	Delay time: Reset error	INTEGER_32	-					
0x2841	Reset error	UNSIGNED_8	-					
0x284F	Current fault	ARRAY [0..63] OF BYTE	-					
Axis A - controller settings								
0x2900	Speed controller: Parameter							
	1: Speed controller: Gain	UNSIGNED_32	1/100000	P				
	2: Speed controller: Reset time	UNSIGNED_16	1/10	P				
	3: Speed controller: Rate time	UNSIGNED_16	1/100	P				
0x2901	Speed controller: Gain - adaptation	UNSIGNED_16	1/100	P	OSC	Rx		
0x2902	Speed controller: Load value	INTEGER_16	1/10	P	OSC	Rx		
0x2903	Speed: Speed setpoint - filter time	UNSIGNED_16	1/10	P				
0x2904	Speed: Actual speed - filter time	UNSIGNED_16	1/10	P				
0x2910	Moments of inertia							
	1: Moment of inertia: Motor	UNSIGNED_32	1/100	P				
	2: Moment of inertia: Load	UNSIGNED_32	1/100	P				
	3: Moment of inertia: Motor-load coupling	UNSIGNED_8	-	P				
0x2939	Switching frequency	UNSIGNED_8	-	P				
0x2941	Current controller: Feedforward control	UNSIGNED_8	-	P				
0x2942	Current controller: Parameter							
	1: Current controller: Gain	UNSIGNED_32	1/100	P				
	2: Current controller: Reset time	UNSIGNED_32	1/100	P				
0x2943	Motor: Current setpoint - filter time	UNSIGNED_16	1/100	P				
Greyed out = read only								

Index	Name	Data type	Scaling	Attributes				
0x2944	Torque: Notch filter torque setpoint							
	1: Notch filter 1: Frequency	UNSIGNED_16	1/10	P				
	2: Notch filter 1: Bandwidth	UNSIGNED_16	1/10	P				
	3: Notch filter 1: Damping	UNSIGNED_8	-	P				
	4: Notch filter 2: Frequency	UNSIGNED_16	1/10	P				
	5: Notch filter 2: Bandwidth	UNSIGNED_16	1/10	P				
	6: Notch filter 2: Damping	UNSIGNED_8	-	P				
0x2945	Torque: Jerk limitation setpoint	UNSIGNED_16	1/10	P				
0x2946	Cyclic sync torque mode: Speed limitation							
	1: Speed limitation: Upper speed limit	INTEGER_32	$480000/2^{31}$	P	OSC	Rx		
	2: Speed limitation: Lower speed limit	INTEGER_32	$480000/2^{31}$	P	OSC	Rx		
0x2947	Inverter characteristic: Voltage grid points (y)							
	1: IC: y1 = U01 (x = 0.00 %)	UNSIGNED_16	1/100	P				
	2: IC: y2 = U02 (x = 6.25 %)	UNSIGNED_16	1/100	P				
	3: IC: y3 = U03 (x = 12.50 %)	UNSIGNED_16	1/100	P				
	4: IC: y4 = U04 (x = 18.75 %)	UNSIGNED_16	1/100	P				
	5: IC: y5 = U05 (x = 25.00 %)	UNSIGNED_16	1/100	P				
	6: IC: y6 = U06 (x = 31.25 %)	UNSIGNED_16	1/100	P				
	7: IC: y7 = U07 (x = 37.50 %)	UNSIGNED_16	1/100	P				
	8: IC: y8 = U08 (x = 42.75 %)	UNSIGNED_16	1/100	P				
	9: IC: y9 = U09 (x = 50.00 %)	UNSIGNED_16	1/100	P				
	10: IC: y10 = U10 (x = 56.25 %)	UNSIGNED_16	1/100	P				
	11: IC: y11 = U11 (x = 62.50 %)	UNSIGNED_16	1/100	P				
	12: IC: y12 = U12 (x = 68.75 %)	UNSIGNED_16	1/100	P				
	13: IC: y13 = U13 (x = 75.00 %)	UNSIGNED_16	1/100	P				
	14: IC: y14 = U14 (x = 81.25 %)	UNSIGNED_16	1/100	P				
	15: IC: y15 = U15 (x = 87.50 %)	UNSIGNED_16	1/100	P				
	16: IC: y16 = U16 (x = 93.25 %)	UNSIGNED_16	1/100	P				
	17: IC: y17 = U17 (x = 100.00 %)	UNSIGNED_16	1/100	P				
0x2980	Position controller: Gain	UNSIGNED_32	1/100	P				
0x2981	Position controller: Gain - adaption	UNSIGNED_16	1/100	P	OSC	Rx		
0x2982	Position controller: Output signal limitation	UNSIGNED_32	$480000/2^{31}$	P	OSC	Rx		
0x2983	Position: Select new actual position	INTEGER_32	-	P	OSC	Rx		
0x2984	Determine target position: Mode	UNSIGNED_8	-	P				
0x29C0	Field controller: Parameter							
	1: Field controller: Gain	UNSIGNED_32	1/100	P				
	2: Field controller: Reset time	UNSIGNED_16	1/10	P				
0x29E0	Field weakening controller: Parameter							
	1: Field weakening controller: Gain	UNSIGNED_32	1/1000	P				
	2: Field weakening controller: Reset time	UNSIGNED_32	1/10	P				
0x29E1	Field set value limitation	UNSIGNED_16	1/100	P	OSC	Rx		
0x29E2	DC link circuit voltage: Filter time	UNSIGNED_16	1/10	P				
0x29E3	Motor voltage act. value: Filter time	UNSIGNED_16	1/10	P				
0x29E4	Voltage reserve range	UNSIGNED_8	-	P				
Axis A - V/f operation								
0x2B00	VFC: V/f characteristic - shape	UNSIGNED_8	-	P				CINH
0x2B01	VFC: V/f characteristic - define reference point							
	1: VFC: V/f characteristic - voltage at reference point	UNSIGNED_16	-	P				
	2: VFC: V/f characteristic - frequency at reference point	UNSIGNED_16	-	P				
Greyed out = read only								

Index	Name	Data type	Scaling	Attributes				
0x2B02	VFC: User-definable V/f characteristic - frequency grid points (x)							
	1: V/f: x1 = f01	INTEGER_16	-	P				
	2: V/f: x2 = f02	INTEGER_16	-	P				
	3: V/f: x3 = f03	INTEGER_16	-	P				
	4: V/f: x4 = f04	INTEGER_16	-	P				
	5: V/f: x5 = f05	INTEGER_16	-	P				
	6: V/f: x6 = f06	INTEGER_16	-	P				
	7: V/f: x7 = f07	INTEGER_16	-	P				
	8: V/f: x8 = f08	INTEGER_16	-	P				
	9: V/f: x9 = f09	INTEGER_16	-	P				
	10: V/f: x10 = f10	INTEGER_16	-	P				
	11: V/f: x11 = f11	INTEGER_16	-	P				
0x2B03	VFC: User-definable V/f characteristic - voltage grid points (y)							
	1: V/f: y1 = U01 (x = f01)	UNSIGNED_32	1/100	P				
	2: V/f: y2 = U02 (x = f02)	UNSIGNED_32	1/100	P				
	3: V/f: y3 = U03 (x = f03)	UNSIGNED_32	1/100	P				
	4: V/f: y4 = U04 (x = f04)	UNSIGNED_32	1/100	P				
	5: V/f: y5 = U05 (x = f05)	UNSIGNED_32	1/100	P				
	6: V/f: y6 = U06 (x = f06)	UNSIGNED_32	1/100	P				
	7: V/f: y7 = U07 (x = f07)	UNSIGNED_32	1/100	P				
	8: V/f: y8 = U08 (x = f08)	UNSIGNED_32	1/100	P				
	9: V/f: y9 = U09 (x = f09)	UNSIGNED_32	1/100	P				
	10: V/f: y10 = U10 (x = f10)	UNSIGNED_32	1/100	P				
	11: V/f: y11 = U11 (x = f11)	UNSIGNED_32	1/100	P				
0x2B04	VFC: Voltage vector control - current setpoint	UNSIGNED_32	1/100	P				
0x2B05	VFC: Voltage vector control parameter							
	1: VFC: Voltage vector controller - gain	UNSIGNED_32	1/100	P				
	2: VFC: Voltage vector controller - reset time	UNSIGNED_32	1/100	P				
0x2B06	VFC: Voltage boost	UNSIGNED_16	1/10	P				
0x2B07	VFC: Load adjustment - parameter							
	1: VFC: Load adjustment - direction of rotation	UNSIGNED_8	-	P				CINH
	2: VFC: Load adjustment - value	UNSIGNED_32	1/100	P				
0x2B08	VFC: Imax controller - parameter							
	1: VFC: Imax controller - gain	UNSIGNED_32	1/1000	P				
	2: VFC: Imax controller - reset time	UNSIGNED_32	1/10	P				
0x2B09	VFC: Slip compensation - parameter							
	1: VFC: Slip compensation - influence	INTEGER_16	1/100	P				
	2: VFC: Slip compensation - filter time	UNSIGNED_16	-	P				
0x2B0A	VFC: Oscillation damping - parameter							
	1: VFC: Oscillation damping - gain	INTEGER_16	-	P				
	2: VFC: Oscillation damping - filter time	UNSIGNED_16	-	P				
	3: VFC: Oscillation damping - limitation	UNSIGNED_16	1/10	P				
	4: VFC: Oscillation damping - final ramp frequency	UNSIGNED_8	-	P				
0x2B0B	VFC: Frequency setpoint	INTEGER_16	1/10		OSC		Tx	
0x2B0C	VFC: Override point of field weakening	INTEGER_16	1/10	P				
0x2B80	DC-injection braking: Current	UNSIGNED_16	1/100	P				
0x2BA0	Flying restart: Activate	UNSIGNED_8	-	P				
0x2BA1	Flying restart: Current	UNSIGNED_16	-	P				
0x2BA2	Flying restart: Start frequency	INTEGER_16	1/10	P				
0x2BA3	Flying restart: Integration time	UNSIGNED_16	-	P				
0x2BA4	Flying restart: Min. deviation	UNSIGNED_16	1/100	P				
Greyed out = read only								

Index	Name	Data type	Scaling	Attributes				
0x2BA5	Flying restart: Delay time	UNSIGNED_16	-	P				
0x2BA6	Flying restart: Result							
	1: Flying restart: Determined speed [rpm]	INTEGER_16	-		OSC		Tx	
	2: Flying restart: Determined speed [n unit]	INTEGER_32	$480000/2^{31}$		OSC		Tx	
Axis A - motor settings								
0x2C00	Motor control	UNSIGNED_8	-	P				CINH
0x2C01	Motor: Common parameters							
	1: Motor: Number of pole pairs	UNSIGNED_8	-					
	2: Motor: Stator resistance	UNSIGNED_32	1/10000	P				
	3: Motor: Stator leakage inductance	UNSIGNED_32	1/1000	P				
	4: Motor: Rated speed	UNSIGNED_16	-	P				
	5: Motor: Rated frequency	UNSIGNED_16	1/10	P				
	6: Motor: Rated power	UNSIGNED_16	1/100	P				
	7: Motor: Rated voltage	UNSIGNED_16	-	P				
	8: Motor: Rated cosine phi	UNSIGNED_16	1/100	P				
	9: Motor: Insulation class	UNSIGNED_8	-	P				
	10: Motor: Designation	STRING(50)	-	P				
0x2C02	Motor (ASM): Parameter							
	1: Motor (ASM): Rotor resistance	UNSIGNED_32	1/10000	P				
	2: Motor (ASM): Mutual inductance	UNSIGNED_32	1/10	P				
	3: Motor (ASM): Magnetising current	UNSIGNED_16	1/100	P				
0x2C03	Motor (SM): Parameter							
	1: Motor (SM): e.m.f. constant (KELL)	UNSIGNED_32	1/10	P				
	2: Motor (SM): Pole position	INTEGER_16	1/10	P				
	3: Motor (SM): Temperature coefficient - magnets (kTN)	INTEGER_16	1/1000	P				
	4: Motor (SM): Pole position	INTEGER_16	1/10	P				
0x2C04	Motor: Lss saturation characteristic - inductance grid points (y)							
	1: Lss: y1 = L01 (x = 0.00 %)	UNSIGNED_16	-	P				
	2: Lss: y2 = L02 (x = 6.25 %)	UNSIGNED_16	-	P				
	3: Lss: y3 = L03 (x = 12.50 %)	UNSIGNED_16	-	P				
	4: Lss: y4 = L04 (x = 18.75 %)	UNSIGNED_16	-	P				
	5: Lss: y5 = L05 (x = 25.00 %)	UNSIGNED_16	-	P				
	6: Lss: y6 = L06 (x = 31.25 %)	UNSIGNED_16	-	P				
	7: Lss: y7 = L07 (x = 37.50 %)	UNSIGNED_16	-	P				
	8: Lss: y8 = L08 (x = 42.75 %)	UNSIGNED_16	-	P				
	9: Lss: y9 = L09 (x = 50.00 %)	UNSIGNED_16	-	P				
	10: Lss: y10 = L10 (x = 56.25 %)	UNSIGNED_16	-	P				
	11: Lss: y11 = L11 (x = 62.50 %)	UNSIGNED_16	-	P				
	12: Lss: y12 = L12 (x = 68.75 %)	UNSIGNED_16	-	P				
	13: Lss: y13 = L13 (x = 75.00 %)	UNSIGNED_16	-	P				
	14: Lss: y14 = L14 (x = 81.25 %)	UNSIGNED_16	-	P				
	15: Lss: y15 = L15 (x = 87.50 %)	UNSIGNED_16	-	P				
	16: Lss: y16 = L16 (x = 93.25 %)	UNSIGNED_16	-	P				
	17: Lss: y17 = L17 (x = 100.00 %)	UNSIGNED_16	-	P				
	18: Motor: Lss saturation characteristic - activation	UNSIGNED_16	-	P				
0x2C05	Motor: Lss saturation charcteristic - reference for current grid points (x)	UNSIGNED_16	1/10	P				
Greyed out = read only								

Index	Name	Data type	Scaling	Attributes				
0x2C06	Motor (SM): Magnet characteristic (current) - grid points							
	1: Magnet characteristic: x1 = i01/iN	UNSIGNED_16	-	P				
	2: Magnet characteristic: y1 = kT01/kTN	UNSIGNED_16	-	P				
	3: Magnet characteristic: x2 = i02/iN	UNSIGNED_16	-	P				
	4: Magnet characteristic: y2 = kT02/kTN	UNSIGNED_16	-	P				
	5: Magnet characteristic: x3 = i03/iN	UNSIGNED_16	-	P				
	6: Magnet characteristic: y3 = kT03/kTN	UNSIGNED_16	-	P				
	7: Magnet characteristic: x4 = i04/iN	UNSIGNED_16	-	P				
	8: Magnet characteristic: y4 = kT04/kTN	UNSIGNED_16	-	P				
0x2C07	Motor (ASM): Lh saturation characteristic - inductance grid points (y)							
	1: Lh: y1 = L01 (x = 0.00 %)	UNSIGNED_16	-	P				
	2: Lh: y2 = L02 (x = 6.25 %)	UNSIGNED_16	-	P				
	3: Lh: y3 = L03 (x = 12.50 %)	UNSIGNED_16	-	P				
	4: Lh: y4 = L04 (x = 18.75 %)	UNSIGNED_16	-	P				
	5: Lh: y5 = L05 (x = 25.00 %)	UNSIGNED_16	-	P				
	6: Lh: y6 = L06 (x = 31.25 %)	UNSIGNED_16	-	P				
	7: Lh: y7 = L07 (x = 37.50 %)	UNSIGNED_16	-	P				
	8: Lh: y8 = L08 (x = 43.75 %)	UNSIGNED_16	-	P				
	9: Lh: y9 = L09 (x = 50.00 %)	UNSIGNED_16	-	P				
	10: Lh: y10 = L10 (x = 56.25 %)	UNSIGNED_16	-	P				
	11: Lh: y11 = L11 (x = 62.50 %)	UNSIGNED_16	-	P				
	12: Lh: y12 = L12 (x = 68.75 %)	UNSIGNED_16	-	P				
	13: Lh: y13 = L13 (x = 75.00 %)	UNSIGNED_16	-	P				
	14: Lh: y14 = L14 (x = 81.25 %)	UNSIGNED_16	-	P				
	15: Lh: y15 = L15 (x = 87.50 %)	UNSIGNED_16	-	P				
	16: Lh: y16 = L16 (x = 93.75 %)	UNSIGNED_16	-	P				
	17: Lh: y17 = L17 (x = 100.00 %)	UNSIGNED_16	-	P				
0x2C08	Motor: Motor parameter setting method	UNSIGNED_8	-	P				
Axis A - feedback system								
0x2C40	Encoder: Type	UNSIGNED_8	-	P				CINH
0x2C41	Hiperface: Parameter							
	1: Hiperface: Determined type code	UNSIGNED_8	-					
	2: Hiperface: User def. encoder - type code	UNSIGNED_8	-	P				CINH
	3: Hiperface: User def. encoder - specifiable revolutions	UNSIGNED_16	-	P				CINH
	4: Hiperface absolute value fault: Response	UNSIGNED_8	-	P				
	5: Hiperface: Serial number	STRING(50)	-					
	6: Hiperface: Raw data - Actual position	UNSIGNED_32	-		OSC		Tx	
	7: Hiperface: Detected Increments / revolution	UNSIGNED_16	-				Tx	
	8: Hiperface: Type code supported by firmware	UNSIGNED_8	-				Tx	
	9: Hiperface: Encoder type	UNSIGNED_8	-				Tx	
	10: Hiperface: Period length of linear encoders	UNSIGNED_32	-				Tx	
0x2C42	Encoder: Parameter							
	1: Encoder: Increments / revolution	UNSIGNED_32	-	P				CINH
	2: Encoder: Supply voltage	UNSIGNED_8	1/10	P				CINH
	3: Encoder: Angle drift - Actual angle error	INTEGER_16	1/10		OSC		Tx	
	4: Encoder: Signal quality - Actual amplitude	UNSIGNED_8	-		OSC			
0x2C43	Resolver: Number of pole pairs	UNSIGNED_8	-	P				CINH
0x2C44	Resolver error compensation: Parameter							
	1: Resolver error compensation: Angle	INTEGER_16	-	P				
	2: Resolver error compensation: Cosine track gain	UNSIGNED_16	-	P				
	3: Resolver error compensation: Sine track gain	UNSIGNED_16	-	P				
Greyed out = read only								

Index	Name	Data type	Scaling	Attributes				
0x2C45	Open circuit in feedback system: Response	UNSIGNED_8	-	P				
0x2C46	Feedback system: Specifiable number of revolutions	UNSIGNED_16	-					
0x2C5F	Feedback system: Parameter CRC	UNSIGNED_32	-					
Axis A - pole position identification								
0x2C60	Monitoring pole position identification: Response	UNSIGNED_8	-	P				
0x2C61	Pole position identification PPI (360°)							
	1: PPI (360°): Current amplitude	UNSIGNED_16	-	P				CINH
	2: PPI (360°): Ramp time	UNSIGNED_16	-	P				CINH
	3: PPI (360°): Direction of rotation	UNSIGNED_8	-	P				CINH
	4: PPI (360°): Fault tolerance	UNSIGNED_8	-	P				
	5: PPI (360°): Absolute current amplitude	UNSIGNED_32	1/100					
0x2C62	Pole position identification PPI (min. movement)							
	1: PPI (min. movement): Current amplitude	UNSIGNED_16	-	P				CINH
	2: PPI (min. movement): Ramp time - current	UNSIGNED_16	-	P				CINH
	3: PPI (min. movement): Gain	UNSIGNED_16	-	P				
	4: PPI (min. movement): Reset time	UNSIGNED_16	1/10	P				
	5: PPI (min. movement): Max. move permitted	UNSIGNED_8	-	P				
	6: PPI (min. movement): Absolute current amplitude	UNSIGNED_32	1/100					
0x2C63	Pole position identification PPI (without movement)							
	1: Behaviour after switch-on	UNSIGNED_8	-	P				CINH
Axis A - touch probe								
0x2D00	Touch probe (TP): Dead time compensation							
	1: Dead time compensation: TP1 delay time	UNSIGNED_16	1/1000	P				
	2: Dead time compensation: TP2 delay time	UNSIGNED_16	1/1000	P				
0x2D01	Touch probe (TP): Time stamp							
	1: TP1: Time stamp - rising edge	UNSIGNED_32	-		OSC		Tx	
	2: TP1: Time stamp - falling edge	UNSIGNED_32	-		OSC		Tx	
	3: TP2: Time stamp - rising edge	UNSIGNED_32	-		OSC		Tx	
	4: TP2: Time stamp - falling edge	UNSIGNED_32	-		OSC		Tx	
Axis A - monitoring functions								
0x2D40	Ixt utilisation							
	1: Power stage utilisation (Ixt): Actual utilisation	UNSIGNED_16	-		OSC		Tx	
	2: Power stage utilisation (Ixt): Warning threshold	UNSIGNED_16	-	P				
	3: Power stage utilisation (Ixt): Error threshold	UNSIGNED_16	-					
	4: Device utilisation (Ixt): Actual utilisation	UNSIGNED_16	-		OSC		Tx	
	5: Device utilisation (Ixt): Warning threshold	UNSIGNED_16	-	P				
	6: Device utilisation (Ixt): Error limit	UNSIGNED_16	-					
0x2D44	Motor speed monitoring							
	1: Motor speed monitoring: Threshold	UNSIGNED_16	-	P				
	2: Motor speed monitoring: Response	UNSIGNED_8	-	P				
0x2D45	Motor phase failure detection							
	1: Motor phase failure 1: Response	UNSIGNED_8	-	P				
	2: Motor phase failure 1: Current threshold	UNSIGNED_8	1/10	P				
	3: Motor phase failure 1: Voltage threshold	UNSIGNED_16	1/10	P				
	4: Motor phase failure 2: Response	UNSIGNED_8	-	P				
0x2D46	Monitoring: Ultimate motor current							
	1: Ultimate motor current: Threshold	UNSIGNED_16	1/10	P				
	2: Ultimate motor current: Response	UNSIGNED_8	-	P				
Greyed out = read only								

Index	Name	Data type	Scaling	Attributes				
0x2D49	Motor temperature monitoring: Parameter							
	1: Motor temperature monitoring: Sensor type	UNSIGNED_8	-	P				
	2: Motor temperature monitoring: Response	UNSIGNED_8	-	P				
	3: Motor temperature monitoring: Warning threshold	INTEGER_16	1/10	P				
	4: Motor temperature monitoring: Error threshold	INTEGER_16	1/10	P				
	5: Motor temperature monitoring: Actual motor temperature	INTEGER_16	1/10		OSC		Tx	
	6: Thermal sensor characteristic: Grid point 1 - temperature	INTEGER_16	1/10	P				
	7: Thermal sensor characteristic: Grid point 2 - temperature	INTEGER_16	1/10	P				
	8: Thermal sensor characteristic: grid point 1 - resistance	INTEGER_16	-	P				
	9: Thermal sensor characteristic: grid point 2 - resistance	INTEGER_16	-	P				
0x2D4C	Motor utilisation (I^2xt): Parameter for the thermal model							
	1: Motor utilisation (I^2xt): Thermal time constant - winding	UNSIGNED_16	-	P				
	2: Motor utilisation (I^2xt): Thermal time constant - laminations	UNSIGNED_16	-	P				
	3: Motor utilisation (I^2xt): Influence of the winding	UNSIGNED_8	-	P				
	4: Motor utilisation (I^2xt): Starting value	UNSIGNED_16	-	P				
0x2D4D	Motor utilisation (I^2xt): User-definable characteristic							
	1: I^2xt: x1 = n01/nN (n01 ~ 0)	UNSIGNED_16	-	P				
	2: I^2xt: y1 = i01/iN (x = n01 ~ 0)	UNSIGNED_16	-	P				
	3: I^2xt: x2 = n02/nN (n02 = limit - reduced cooling)	UNSIGNED_16	-	P				
	4: I^2xt: y2 = i02/iN (x = n02 = limit - reduced cooling)	UNSIGNED_16	-	P				
	5: I^2xt: x3 = n03/nN (n03 = rated speed)	UNSIGNED_16	-	P				
	6: I^2xt: y3 = i03/iN (x = n03 = rated speed)	UNSIGNED_16	-	P				
	7: I^2xt: x4 = n04/nN (n04 = limit - field weakening)	UNSIGNED_16	-	P				
	8: I^2xt: y4 = i04/iN (x = n04 = limit - field weakening)	UNSIGNED_16	-	P				
0x2D4E	Motor utilisation (I^2xt): Motor overload warning threshold	UNSIGNED_16	-	P				
0x2D4F	Motor utilisation (I^2xt): Actual utilisation	UNSIGNED_16	-		OSC		Tx	
0x2D50	Motor utilisation (I^2xt): Motor overload error							
	1: Motor utilisation (I^2xt): Response	UNSIGNED_8	-	P				
	2: Motor utilisation (I^2xt): Error threshold	UNSIGNED_16	-	P				
0x2D81	Counter: Operating time							
	1: Device: Operating time	UNSIGNED_32	-					
	2: Device: Power-on time	UNSIGNED_32	-					
0x2D82	Motor: Actual voltage - Veff, phase-phase	UNSIGNED_32	1/10		OSC		Tx	
0x2D83	Motor: Phase currents							
	1: Zero system current	INTEGER_32	1/100		OSC		Tx	
	2: Current - phase U	INTEGER_32	1/100		OSC		Tx	
	3: Current - phase V	INTEGER_32	1/100		OSC		Tx	
	4: Current - phase W	INTEGER_32	1/100		OSC		Tx	
0x2D84	Heatsink temperature							
	1: Heatsink temperature: Actual temperature	INTEGER_16	1/10		OSC			
	2: Heatsink temperature: Warning threshold	INTEGER_16	1/10	P				
	3: Heatsink temperature: Threshold - switch-on fan	INTEGER_16	1/10	P				
	4: Heatsink temperature: Threshold - switch-off fan	INTEGER_16	1/10	P				
0x2D8A	Speed monitoring: Actual speed error	INTEGER_32	-		OSC		Tx	
0x2DD0	Field: Values							
	1: Field: Actual field	UNSIGNED_16	-		OSC		Tx	
	2: Field: Setpoint field	UNSIGNED_16	-		OSC		Tx	
Greyed out = read only								

Index	Name	Data type	Scaling	Attributes				
0x2DD1	Motor: Currents							
	1: D current (id): Actual D current	INTEGER_16	1/100		OSC		Tx	
	2: Q current (iq): Actual Q current	INTEGER_16	1/100		OSC		Tx	
	3: D current (id): Setpoint D current	INTEGER_16	1/100		OSC		Tx	
	4: Q current (iq): Setpoint Q current	INTEGER_16	1/100		OSC		Tx	
	5: Motor current Ieff	INTEGER_16	1/100		OSC		Tx	
0x2DD2	Position: Target position interpolated	INTEGER_32	-		OSC		Tx	
0x2DD3	Target speeds							
	1: Speed: Setpoint speed	INTEGER_32	-		OSC		Tx	
	2: Speed: Setpoint speed 2	INTEGER_32	-		OSC		Tx	
	3: Speed: Limited setpoint speed	INTEGER_32	-		OSC		Tx	
0x2DD4	Speed controller: Output signal							
	1: Speed controller: Output signal 1	INTEGER_16	1/10		OSC		Tx	
	2: Speed controller: Output signal 2	INTEGER_16	1/10		OSC		Tx	
0x2DD5	Target torque	INTEGER_32	1/100		OSC		Tx	
0x2DD6	Torque: Filter cascade							
	1: Torque: Filter cascade - starting value	INTEGER_16	1/10		OSC		Tx	
	2: Torque: Notch filter 1 - input value	INTEGER_16	1/10		OSC		Tx	
	3: Torque: Notch filter 2 - input value	INTEGER_16	1/10		OSC		Tx	
	4: Torque: Filtered setpoint torque	INTEGER_16	1/10		OSC		Tx	
0x2DD7	Voltage values							
	1: Actual motor voltage limit: Actual voltage	INTEGER_16	1/10		OSC		Tx	
	2: D-current controller: Output signal	INTEGER_16	1/10		OSC		Tx	
	3: Q-current controller: Output signal	INTEGER_16	1/10		OSC		Tx	
	4: D-voltage (magnetisation)	INTEGER_16	1/10		OSC		Tx	
	5: Q-voltage (torque)	INTEGER_16	1/10		OSC		Tx	
0x2DDC	Slip: Actual slip	INTEGER_16	1/10		OSC		Tx	
0x2DDD	Device: Actual output frequency	INTEGER_16	1/10		OSC		Tx	
0x2DDE	Motor: Actual position of rotor angle	INTEGER_16	-		OSC		Tx	
0x2DDF	Axis: Device data							
	1: Axis: Rated current	UNSIGNED_16	1/100		OSC		Tx	
	2: Axis: Maximum current	UNSIGNED_16	1/100		OSC		Tx	
	5: Axis: Supported feedback type	UNSIGNED_8	-				Tx	
Axis B - holding brake control								
0x3020	Brake control: Settings							
	1: Brake: Mode	UNSIGNED_8	-	P		Rx		
	2: Brake: Application time	UNSIGNED_16	-	P				
	3: Brake: Release time	UNSIGNED_16	-	P				
	4: Brake: Brake recognition	UNSIGNED_16	-					
	5: Brake: Control logic	UNSIGNED_8	-	P				
	6: Brake monitoring: Response	UNSIGNED_8	-	P				
Axis B - device control								
0x3022	Axis command	UNSIGNED_32	-					
0x3023	Axis command: Progress	UNSIGNED_8	-					
0x3024	Device control via PDO: Activation	UNSIGNED_8	-	P				CINH
0x3025	Mode: Selection	UNSIGNED_8	-	P				CINH
0x3026	Quick stop: Duration in case of trouble	UNSIGNED_32	-	P				
0x3030	Lenze control word	UNSIGNED_16	-		OSC	Rx		
0x3031	Lenze status word	UNSIGNED_16	-		OSC		Tx	
0x3032	Identification: Status word	UNSIGNED_16	-		OSC			
0x3033	Lenze statusword 2	UNSIGNED_16	-		OSC		Tx	
Greyed out = read only								

Index	Name	Data type	Scaling	Attributes				
0x3035	Manual test mode: Settings							
	1: Manual test mode: Setpoint current	INTEGER_16	-		OSC	Rx		
	2: Manual test mode: Frequency	INTEGER_16	1/10		OSC	Rx		
	3: Manual test mode: Starting angle	INTEGER_16	1/10					
0x3036	Manual jog: Settings							
	1: Manual jog: Setpoint current	UNSIGNED_16	-			Rx		
	2: Manual jog: Frequency	INTEGER_16	1/10		OSC	Rx		
	3: Manual jog: Ramp time - current	UNSIGNED_16	-	P				
	4: Manual jog: Ramp time - frequency	UNSIGNED_16	-	P				
	5: Manual jog: Time monitoring	UNSIGNED_32	-	P				
	6: Manual jog: Current controller gain	UNSIGNED_32	1/100	P				
	7: Manual jog: Current controller reset time	UNSIGNED_32	1/100	P				
Axis B - error management								
0x3040	Delay time: Reset error	INTEGER_32	-					
0x3041	Reset error	UNSIGNED_8	-					
0x304F	Current fault	ARRAY [0..63] OF BYTE	-					
Axis B - controller settings								
0x3100	Speed controller: Parameter							
	1: Speed controller: Gain	UNSIGNED_32	1/100000	P				
	2: Speed controller: Reset time	UNSIGNED_16	1/10	P				
	3: Speed controller: Rate time	UNSIGNED_16	1/100	P				
0x3101	Speed controller: Gain - adaption	UNSIGNED_16	1/100	P	OSC	Rx		
0x3102	Speed controller: Load value	INTEGER_16	1/10	P	OSC	Rx		
0x3103	Speed: Speed setpoint - filter time	UNSIGNED_16	1/10	P				
0x3104	Speed: Actual speed - filter time	UNSIGNED_16	1/10	P				
0x3110	Moments of inertia							
	1: Moment of inertia: Motor	UNSIGNED_32	1/100	P				
	2: Moment of inertia: Load	UNSIGNED_32	1/100	P				
	3: Moment of inertia: Motor-load coupling	UNSIGNED_8	-	P				
0x3139	Switching frequency	UNSIGNED_8	-	P				
0x3141	Current controller: Feedforward control	UNSIGNED_8	-	P				
0x3142	Current controller: Parameter							
	1: Current controller: Gain	UNSIGNED_32	1/100	P				
	2: Current controller: Reset time	UNSIGNED_32	1/100	P				
0x3143	Motor: Current setpoint - filter time	UNSIGNED_16	1/100	P				
0x3144	Torque: Notch filter torque setpoint							
	1: Notch filter 1: Frequency	UNSIGNED_16	1/10	P				
	2: Notch filter 1: Bandwidth	UNSIGNED_16	1/10	P				
	3: Notch filter 1: Damping	UNSIGNED_8	-	P				
	4: Notch filter 2: Frequency	UNSIGNED_16	1/10	P				
	5: Notch filter 2: Bandwidth	UNSIGNED_16	1/10	P				
	6: Notch filter 2: Damping	UNSIGNED_8	-	P				
0x3145	Torque: Jerk limitation setpoint	UNSIGNED_16	1/10	P				
0x3146	Cyclic sync torque mode: Speed limitation							
	1: Speed limitation: Upper speed limit	INTEGER_32	$480000/2^{31}$	P	OSC	Rx		
	2: Speed limitation: Lower speed limit	INTEGER_32	$480000/2^{31}$	P	OSC	Rx		
Greyed out = read only								

Index	Name	Data type	Scaling	Attributes				
0x3147	Inverter characteristic: Voltage grid points (y)							
	1: IC: y1 = U01 (x = 0.00 %)	UNSIGNED_16	1/100	P				
	2: IC: y2 = U02 (x = 6.25 %)	UNSIGNED_16	1/100	P				
	3: IC: y3 = U03 (x = 12.50 %)	UNSIGNED_16	1/100	P				
	4: IC: y4 = U04 (x = 18.75 %)	UNSIGNED_16	1/100	P				
	5: IC: y5 = U05 (x = 25.00 %)	UNSIGNED_16	1/100	P				
	6: IC: y6 = U06 (x = 31.25 %)	UNSIGNED_16	1/100	P				
	7: IC: y7 = U07 (x = 37.50 %)	UNSIGNED_16	1/100	P				
	8: IC: y8 = U08 (x = 42.75 %)	UNSIGNED_16	1/100	P				
	9: IC: y9 = U09 (x = 50.00 %)	UNSIGNED_16	1/100	P				
	10: IC: y10 = U10 (x = 56.25 %)	UNSIGNED_16	1/100	P				
	11: IC: y11 = U11 (x = 62.50 %)	UNSIGNED_16	1/100	P				
	12: IC: y12 = U12 (x = 68.75 %)	UNSIGNED_16	1/100	P				
	13: IC: y13 = U13 (x = 75.00 %)	UNSIGNED_16	1/100	P				
	14: IC: y14 = U14 (x = 81.25 %)	UNSIGNED_16	1/100	P				
	15: IC: y15 = U15 (x = 87.50 %)	UNSIGNED_16	1/100	P				
	16: IC: y16 = U16 (x = 93.25 %)	UNSIGNED_16	1/100	P				
	17: IC: y17 = U17 (x = 100.00 %)	UNSIGNED_16	1/100	P				
0x3180	Position controller: Gain	UNSIGNED_32	1/100	P				
0x3181	Position controller: Gain - adaption	UNSIGNED_16	1/100	P	OSC	Rx		
0x3182	Position controller: Output signal limitation	UNSIGNED_32	$480000/2^{31}$	P	OSC	Rx		
0x3183	Position: Select new actual position	INTEGER_32	-	P	OSC	Rx		
0x3184	Determine target position: Mode	UNSIGNED_8	-	P				
0x31C0	Field controller: Parameter							
	1: Field controller: Gain	UNSIGNED_32	1/100	P				
	2: Field controller: Reset time	UNSIGNED_16	1/10	P				
0x31E0	Field weakening controller: Parameter							
	1: Field weakening controller: Gain	UNSIGNED_32	1/1000	P				
	2: Field weakening controller: Reset time	UNSIGNED_32	1/10	P				
0x31E1	Field set value limitation	UNSIGNED_16	1/100	P	OSC	Rx		
0x31E2	DC link circuit voltage: Filter time	UNSIGNED_16	1/10	P				
0x31E3	Motor voltage act. value: Filter time	UNSIGNED_16	1/10	P				
0x31E4	Voltage reserve range	UNSIGNED_8	-	P				
Axis B - V/f operation								
0x3300	VFC: V/f characteristic - shape	UNSIGNED_8	-	P				CINH
0x3301	VFC: V/f characteristic - define reference point							
	1: VFC: V/f characteristic - voltage at reference point	UNSIGNED_16	-	P				
	2: VFC: V/f characteristic - frequency at reference point	UNSIGNED_16	-	P				
0x3302	VFC: User-definable V/f characteristic - frequency grid points (x)							
	1: V/f: x1 = f01	INTEGER_16	-	P				
	2: V/f: x2 = f02	INTEGER_16	-	P				
	3: V/f: x3 = f03	INTEGER_16	-	P				
	4: V/f: x4 = f04	INTEGER_16	-	P				
	5: V/f: x5 = f05	INTEGER_16	-	P				
	6: V/f: x6 = f06	INTEGER_16	-	P				
	7: V/f: x7 = f07	INTEGER_16	-	P				
	8: V/f: x8 = f08	INTEGER_16	-	P				
	9: V/f: x9 = f09	INTEGER_16	-	P				
	10: V/f: x10 = f10	INTEGER_16	-	P				
	11: V/f: x11 = f11	INTEGER_16	-	P				
Greyed out = read only								

Index	Name	Data type	Scaling	Attributes				
0x3303	VFC: User-definable V/f characteristic - voltage grid points (y)							
	1: V/f: y1 = U01 (x = f01)	UNSIGNED_32	1/100	P				
	2: V/f: y2 = U02 (x = f02)	UNSIGNED_32	1/100	P				
	3: V/f: y3 = U03 (x = f03)	UNSIGNED_32	1/100	P				
	4: V/f: y4 = U04 (x = f04)	UNSIGNED_32	1/100	P				
	5: V/f: y5 = U05 (x = f05)	UNSIGNED_32	1/100	P				
	6: V/f: y6 = U06 (x = f06)	UNSIGNED_32	1/100	P				
	7: V/f: y7 = U07 (x = f07)	UNSIGNED_32	1/100	P				
	8: V/f: y8 = U08 (x = f08)	UNSIGNED_32	1/100	P				
	9: V/f: y9 = U09 (x = f09)	UNSIGNED_32	1/100	P				
	10: V/f: y10 = U10 (x = f10)	UNSIGNED_32	1/100	P				
	11: V/f: y11 = U11 (x = f11)	UNSIGNED_32	1/100	P				
0x3304	VFC: Voltage vector control - current setpoint	UNSIGNED_32	1/100	P				
0x3305	VFC: Voltage vector control parameter							
	1: VFC: Voltage vector controller - gain	UNSIGNED_32	1/100	P				
	2: VFC: Voltage vector controller - reset time	UNSIGNED_32	1/100	P				
0x3306	VFC: Voltage boost	UNSIGNED_16	1/10	P				
0x3307	VFC: Load adjustment - parameter							
	1: VFC: Load adjustment - direction of rotation	UNSIGNED_8	-	P				CINH
	2: VFC: Load adjustment - value	UNSIGNED_32	1/100	P				
0x3308	VFC: Imax controller - parameter							
	1: VFC: Imax controller - gain	UNSIGNED_32	1/1000	P				
	2: VFC: Imax controller - reset time	UNSIGNED_32	1/10	P				
0x3309	VFC: Slip compensation - parameter							
	1: VFC: Slip compensation - influence	INTEGER_16	1/100	P				
	2: VFC: Slip compensation - filter time	UNSIGNED_16	-	P				
0x330A	VFC: Oscillation damping - parameter							
	1: VFC: Oscillation damping - gain	INTEGER_16	-	P				
	2: VFC: Oscillation damping - filter time	UNSIGNED_16	-	P				
	3: VFC: Oscillation damping - limitation	UNSIGNED_16	1/10	P				
	4: VFC: Oscillation damping - final ramp frequency	UNSIGNED_8	-	P				
0x330B	VFC: Frequency setpoint	INTEGER_16	1/10		OSC		Tx	
0x330C	VFC: Override point of field weakening	INTEGER_16	1/10	P				
0x3380	DC-injection braking: Current	UNSIGNED_16	1/100	P				
0x33A0	Flying restart: Activate	UNSIGNED_8	-	P				
0x33A1	Flying restart: Current	UNSIGNED_16	-	P				
0x33A2	Flying restart: Start frequency	INTEGER_16	1/10	P				
0x33A3	Flying restart: Integration time	UNSIGNED_16	-	P				
0x33A4	Flying restart: Min. deviation	UNSIGNED_16	1/100	P				
0x33A5	Flying restart: Delay time	UNSIGNED_16	-	P				
0x33A6	Flying restart: Result							
	1: Flying restart: Determined speed [rpm]	INTEGER_16	-		OSC		Tx	
	2: Flying restart: Determined speed [n unit]	INTEGER_32	$480000/2^{31}$		OSC		Tx	
Greyed out = read only								

Index	Name	Data type	Scaling	Attributes				
Axis B - motor settings								
0x3400	Motor control	UNSIGNED_8	-	P				CINH
0x3401	Motor: Common parameters							
	1: Motor: Number of pole pairs	UNSIGNED_8	-					
	2: Motor: Stator resistance	UNSIGNED_32	1/10000	P				
	3: Motor: Stator leakage inductance	UNSIGNED_32	1/1000	P				
	4: Motor: Rated speed	UNSIGNED_16	-	P				
	5: Motor: Rated frequency	UNSIGNED_16	1/10	P				
	6: Motor: Rated power	UNSIGNED_16	1/100	P				
	7: Motor: Rated voltage	UNSIGNED_16	-	P				
	8: Motor: Rated cosine phi	UNSIGNED_16	1/100	P				
	9: Motor: Insulation class	UNSIGNED_8	-	P				
	10: Motor: Designation	STRING(50)	-	P				
0x3402	Motor (ASM): Parameter							
	1: Motor (ASM): Rotor resistance	UNSIGNED_32	1/10000	P				
	2: Motor (ASM): Mutual inductance	UNSIGNED_32	1/10	P				
	3: Motor (ASM): Magnetising current	UNSIGNED_16	1/100	P				
0x3403	Motor (SM): Parameter							
	1: Motor (SM): e.m.f. constant (KELL)	UNSIGNED_32	1/10	P				
	2: Motor (SM): Pole position	INTEGER_16	1/10	P				
	3: Motor (SM): Temperature coefficient - magnets (kTN)	INTEGER_16	1/1000	P				
	4: Motor (SM): Pole position	INTEGER_16	1/10	P				
0x3404	Motor: Lss saturation characteristic - inductance grid points (y)							
	1: Lss: y1 = L01 (x = 0.00 %)	UNSIGNED_16	-	P				
	2: Lss: y2 = L02 (x = 6.25 %)	UNSIGNED_16	-	P				
	3: Lss: y3 = L03 (x = 12.50 %)	UNSIGNED_16	-	P				
	4: Lss: y4 = L04 (x = 18.75 %)	UNSIGNED_16	-	P				
	5: Lss: y5 = L05 (x = 25.00 %)	UNSIGNED_16	-	P				
	6: Lss: y6 = L06 (x = 31.25 %)	UNSIGNED_16	-	P				
	7: Lss: y7 = L07 (x = 37.50 %)	UNSIGNED_16	-	P				
	8: Lss: y8 = L08 (x = 42.75 %)	UNSIGNED_16	-	P				
	9: Lss: y9 = L09 (x = 50.00 %)	UNSIGNED_16	-	P				
	10: Lss: y10 = L10 (x = 56.25 %)	UNSIGNED_16	-	P				
	11: Lss: y11 = L11 (x = 62.50 %)	UNSIGNED_16	-	P				
	12: Lss: y12 = L12 (x = 68.75 %)	UNSIGNED_16	-	P				
	13: Lss: y13 = L13 (x = 75.00 %)	UNSIGNED_16	-	P				
	14: Lss: y14 = L14 (x = 81.25 %)	UNSIGNED_16	-	P				
	15: Lss: y15 = L15 (x = 87.50 %)	UNSIGNED_16	-	P				
	16: Lss: y16 = L16 (x = 93.25 %)	UNSIGNED_16	-	P				
	17: Lss: y17 = L17 (x = 100.00 %)	UNSIGNED_16	-	P				
	18: Motor: Lss saturation characteristic - activation	UNSIGNED_16	-	P				
0x3405	Motor: Lss saturation charcteristic - reference for current grid points (x)	UNSIGNED_16	1/10	P				
0x3406	Motor (SM): Magnet characteristic (current) - grid points							
	1: Magnet characteristic: x1 = i01/iN	UNSIGNED_16	-	P				
	2: Magnet characteristic: y1 = kT01/kTN	UNSIGNED_16	-	P				
	3: Magnet characteristic: x2 = i02/iN	UNSIGNED_16	-	P				
	4: Magnet characteristic: y2 = kT02/kTN	UNSIGNED_16	-	P				
	5: Magnet characteristic: x3 = i03/iN	UNSIGNED_16	-	P				
	6: Magnet characteristic: y3 = kT03/kTN	UNSIGNED_16	-	P				
	7: Magnet characteristic: x4 = i04/iN	UNSIGNED_16	-	P				
	8: Magnet characteristic: y4 = kT04/kTN	UNSIGNED_16	-	P				
Greyed out = read only								

Index	Name	Data type	Scaling	Attributes				
0x3407	Motor (ASM): Lh saturation characteristic - inductance grid points (y)							
	1: Lh: y1 = L01 (x = 0.00 %)	UNSIGNED_16	-	P				
	2: Lh: y2 = L02 (x = 6.25 %)	UNSIGNED_16	-	P				
	3: Lh: y3 = L03 (x = 12.50 %)	UNSIGNED_16	-	P				
	4: Lh: y4 = L04 (x = 18.75 %)	UNSIGNED_16	-	P				
	5: Lh: y5 = L05 (x = 25.00 %)	UNSIGNED_16	-	P				
	6: Lh: y6 = L06 (x = 31.25 %)	UNSIGNED_16	-	P				
	7: Lh: y7 = L07 (x = 37.50 %)	UNSIGNED_16	-	P				
	8: Lh: y8 = L08 (x = 43.75 %)	UNSIGNED_16	-	P				
	9: Lh: y9 = L09 (x = 50.00 %)	UNSIGNED_16	-	P				
	10: Lh: y10 = L10 (x = 56.25 %)	UNSIGNED_16	-	P				
	11: Lh: y11 = L11 (x = 62.50 %)	UNSIGNED_16	-	P				
	12: Lh: y12 = L12 (x = 68.75 %)	UNSIGNED_16	-	P				
	13: Lh: y13 = L13 (x = 75.00 %)	UNSIGNED_16	-	P				
	14: Lh: y14 = L14 (x = 81.25 %)	UNSIGNED_16	-	P				
	15: Lh: y15 = L15 (x = 87.50 %)	UNSIGNED_16	-	P				
	16: Lh: y16 = L16 (x = 93.75 %)	UNSIGNED_16	-	P				
	17: Lh: y17 = L17 (x = 100.00 %)	UNSIGNED_16	-	P				
0x3408	Motor: Motor parameter setting method	UNSIGNED_8	-	P				
Axis B - feedback system								
0x3440	Encoder: Type	UNSIGNED_8	-	P				CINH
0x3441	Hiperface: Parameter							
	1: Hiperface: Determined type code	UNSIGNED_8	-					
	2: Hiperface: User def. encoder - type code	UNSIGNED_8	-	P				CINH
	3: Hiperface: User def. encoder - specifiable revolutions	UNSIGNED_16	-	P				CINH
	4: Hiperface absolute value fault: Response	UNSIGNED_8	-	P				
	5: Hiperface: Serial number	STRING(50)	-					
	6: Hiperface: Raw data - Actual position	UNSIGNED_32	-		OSC		Tx	
	7: Hiperface: Detected Increments / revolution	UNSIGNED_16	-				Tx	
	8: Hiperface: Type code supported by firmware	UNSIGNED_8	-				Tx	
	9: Hiperface: Encoder type	UNSIGNED_8	-				Tx	
	10: Hiperface: Period length of linear encoders	UNSIGNED_32	-				Tx	
0x3442	Encoder: Parameter							
	1: Encoder: Increments / revolution	UNSIGNED_32	-	P				CINH
	2: Encoder: Supply voltage	UNSIGNED_8	1/10	P				CINH
	3: Encoder: Angle drift - Actual angle error	INTEGER_16	1/10		OSC		Tx	
	4: Encoder: Signal quality - Actual amplitude	UNSIGNED_8	-		OSC			
0x3443	Resolver: Number of pole pairs	UNSIGNED_8	-	P				CINH
0x3444	Resolver error compensation: Parameter							
	1: Resolver error compensation: Angle	INTEGER_16	-	P				
	2: Resolver error compensation: Cosine track gain	UNSIGNED_16	-	P				
	3: Resolver error compensation: Sine track gain	UNSIGNED_16	-	P				
0x3445	Open circuit in feedback system: Response	UNSIGNED_8	-	P				
0x3446	Feedback system: Specifiable number of revolutions	UNSIGNED_16	-					
0x345F	Feedback system: Parameter CRC	UNSIGNED_32	-					
Greyed out = read only								

Index	Name	Data type	Scaling	Attributes				
Axis B - pole position identification								
0x3460	Monitoring pole position identification: Response	UNSIGNED_8	-	P				
0x3461	Pole position identification PPI (360°)							
	1: PPI (360°): Current amplitude	UNSIGNED_16	-	P				CINH
	2: PPI (360°): Ramp time	UNSIGNED_16	-	P				CINH
	3: PPI (360°): Direction of rotation	UNSIGNED_8	-	P				CINH
	4: PPI (360°): Fault tolerance	UNSIGNED_8	-	P				
	5: PPI (360°): Absolute current amplitude	UNSIGNED_32	1/100					
0x3462	Pole position identification PPI (min. movement)							
	1: PPI (min. movement): Current amplitude	UNSIGNED_16	-	P				CINH
	2: PPI (min. movement): Ramp time - current	UNSIGNED_16	-	P				CINH
	3: PPI (min. movement): Gain	UNSIGNED_16	-	P				
	4: PPI (min. movement): Reset time	UNSIGNED_16	1/10	P				
	5: PPI (min. movement): Max. move permitted	UNSIGNED_8	-	P				
	6: PPI (min. movement): Absolute current amplitude	UNSIGNED_32	1/100					
0x3463	Pole position identification PPI (without movement)							
	1: Behaviour after switch-on	UNSIGNED_8	-	P				CINH
Axis B - touch probe								
0x3500	Touch probe (TP): Dead time compensation							
	1: Dead time compensation: TP1 delay time	UNSIGNED_16	1/1000	P				
	2: Dead time compensation: TP2 delay time	UNSIGNED_16	1/1000	P				
0x3501	Touch probe (TP): Time stamp							
	1: TP1: Time stamp - rising edge	UNSIGNED_32	-		OSC		Tx	
	2: TP1: Time stamp - falling edge	UNSIGNED_32	-		OSC		Tx	
	3: TP2: Time stamp - rising edge	UNSIGNED_32	-		OSC		Tx	
	4: TP2: Time stamp - falling edge	UNSIGNED_32	-		OSC		Tx	
Axis B - monitoring functions								
0x3540	Ixt utilisation							
	1: Power stage utilisation (Ixt): Actual utilisation	UNSIGNED_16	-		OSC		Tx	
	2: Power stage utilisation (Ixt): Warning threshold	UNSIGNED_16	-	P				
	3: Power stage utilisation (Ixt): Error threshold	UNSIGNED_16	-					
	4: Device utilisation (Ixt): Actual utilisation	UNSIGNED_16	-		OSC		Tx	
	5: Device utilisation (Ixt): Warning threshold	UNSIGNED_16	-	P				
	6: Device utilisation (Ixt): Error limit	UNSIGNED_16	-					
0x3544	Motor speed monitoring							
	1: Motor speed monitoring: Threshold	UNSIGNED_16	-	P				
	2: Motor speed monitoring: Response	UNSIGNED_8	-	P				
0x3545	Motor phase failure detection							
	1: Motor phase failure 1: Response	UNSIGNED_8	-	P				
	2: Motor phase failure 1: Current threshold	UNSIGNED_8	1/10	P				
	3: Motor phase failure 1: Voltage threshold	UNSIGNED_16	1/10	P				
	4: Motor phase failure 2: Response	UNSIGNED_8	-	P				
0x3546	Monitoring: Ultimate motor current							
	1: Ultimate motor current: Threshold	UNSIGNED_16	1/10	P				
	2: Ultimate motor current: Response	UNSIGNED_8	-	P				
Greyed out = read only								

Index	Name	Data type	Scaling	Attributes				
0x3549	Motor temperature monitoring: Parameter							
	1: Motor temperature monitoring: Sensor type	UNSIGNED_8	-	P				
	2: Motor temperature monitoring: Response	UNSIGNED_8	-	P				
	3: Motor temperature monitoring: Warning threshold	INTEGER_16	1/10	P				
	4: Motor temperature monitoring: Error threshold	INTEGER_16	1/10	P				
	5: Motor temperature monitoring: Actual motor temperature	INTEGER_16	1/10		OSC		Tx	
	6: Thermal sensor characteristic: Grid point 1 - temperature	INTEGER_16	1/10	P				
	7: Thermal sensor characteristic: Grid point 2 - temperature	INTEGER_16	1/10	P				
	8: Thermal sensor characteristic: grid point 1 - resistance	INTEGER_16	-	P				
	9: Thermal sensor characteristic: grid point 2 - resistance	INTEGER_16	-	P				
0x354C	Motor utilisation (I²xt): Parameter for the thermal model							
	1: Motor utilisation (I²xt): Thermal time constant - winding	UNSIGNED_16	-	P				
	2: Motor utilisation (I²xt): Thermal time constant - laminations	UNSIGNED_16	-	P				
	3: Motor utilisation (I²xt): Influence of the winding	UNSIGNED_8	-	P				
	4: Motor utilisation (I²xt): Starting value	UNSIGNED_16	-	P				
0x354D	Motor utilisation (I²xt): User-definable characteristic							
	1: I²xt: x1 = n01/nN (n01 ~ 0)	UNSIGNED_16	-	P				
	2: I²xt: y1 = i01/iN (x = n01 ~ 0)	UNSIGNED_16	-	P				
	3: I²xt: x2 = n02/nN (n02 = limit - reduced cooling)	UNSIGNED_16	-	P				
	4: I²xt: y2 = i02/iN (x = n02 = limit - reduced cooling)	UNSIGNED_16	-	P				
	5: I²xt: x3 = n03/nN (n03 = rated speed)	UNSIGNED_16	-	P				
	6: I²xt: y3 = i03/iN (x = n03 = rated speed)	UNSIGNED_16	-	P				
	7: I²xt: x4 = n04/nN (n04 = limit - field weakening)	UNSIGNED_16	-	P				
	8: I²xt: y4 = i04/iN (x = n04 = limit - field weakening)	UNSIGNED_16	-	P				
0x354E	Motor utilisation (I²xt): Motor overload warning threshold	UNSIGNED_16	-	P				
0x354F	Motor utilisation (I²xt): Actual utilisation	UNSIGNED_16	-		OSC		Tx	
0x3550	Motor utilisation (I²xt): Motor overload error							
	1: Motor utilisation (I²xt): Response	UNSIGNED_8	-	P				
	2: Motor utilisation (I²xt): Error threshold	UNSIGNED_16	-	P				
0x3581	Counter: Operating time							
	1: Device: Operating time	UNSIGNED_32	-					
	2: Device: Power-on time	UNSIGNED_32	-					
0x3582	Motor: Actual voltage - Veff, phase-phase	UNSIGNED_32	1/10		OSC		Tx	
0x3583	Motor: Phase currents							
	1: Zero system current	INTEGER_32	1/100		OSC		Tx	
	2: Current - phase U	INTEGER_32	1/100		OSC		Tx	
	3: Current - phase V	INTEGER_32	1/100		OSC		Tx	
	4: Current - phase W	INTEGER_32	1/100		OSC		Tx	
0x3584	Heatsink temperature							
	1: Heatsink temperature: Actual temperature	INTEGER_16	1/10		OSC			
	2: Heatsink temperature: Warning threshold	INTEGER_16	1/10	P				
	3: Heatsink temperature: Threshold - switch-on fan	INTEGER_16	1/10	P				
	4: Heatsink temperature: Threshold - switch-off fan	INTEGER_16	1/10	P				
0x358A	Speed monitoring: Actual speed error	INTEGER_32	-		OSC		Tx	
0x35D0	Field: Values							
	1: Field: Actual field	UNSIGNED_16	-		OSC		Tx	
	2: Field: Setpoint field	UNSIGNED_16	-		OSC		Tx	
Greyed out = read only								

Index	Name	Data type	Scaling	Attributes				
0x35D1	Motor: Currents							
	1: D current (id): Actual D current	INTEGER_16	1/100		OSC		Tx	
	2: Q current (iq): Actual Q current	INTEGER_16	1/100		OSC		Tx	
	3: D current (id): Setpoint D current	INTEGER_16	1/100		OSC		Tx	
	4: Q current (iq): Setpoint Q current	INTEGER_16	1/100		OSC		Tx	
	5: Motor current Ieff	INTEGER_16	1/100		OSC		Tx	
0x35D2	Position: Target position interpolated	INTEGER_32	-		OSC		Tx	
0x35D3	Target speeds							
	1: Speed: Setpoint speed	INTEGER_32	-		OSC		Tx	
	2: Speed: Setpoint speed 2	INTEGER_32	-		OSC		Tx	
	3: Speed: Limited setpoint speed	INTEGER_32	-		OSC		Tx	
0x35D4	Speed controller: Output signal							
	1: Speed controller: Output signal 1	INTEGER_16	1/10		OSC		Tx	
	2: Speed controller: Output signal 2	INTEGER_16	1/10		OSC		Tx	
0x35D5	Target torque	INTEGER_32	1/100		OSC		Tx	
0x35D6	Torque: Filter cascade							
	1: Torque: Filter cascade - starting value	INTEGER_16	1/10		OSC		Tx	
	2: Torque: Notch filter 1 - input value	INTEGER_16	1/10		OSC		Tx	
	3: Torque: Notch filter 2 - input value	INTEGER_16	1/10		OSC		Tx	
	4: Torque: Filtered setpoint torque	INTEGER_16	1/10		OSC		Tx	
0x35D7	Voltage values							
	1: Actual motor voltage limit: Actual voltage	INTEGER_16	1/10		OSC		Tx	
	2: D-current controller: Output signal	INTEGER_16	1/10		OSC		Tx	
	3: Q-current controller: Output signal	INTEGER_16	1/10		OSC		Tx	
	4: D-voltage (magnetisation)	INTEGER_16	1/10		OSC		Tx	
	5: Q-voltage (torque)	INTEGER_16	1/10		OSC		Tx	
0x35DC	Slip: Actual slip	INTEGER_16	1/10		OSC		Tx	
0x35DD	Device: Actual output frequency	INTEGER_16	1/10		OSC		Tx	
0x35DE	Motor: Actual position of rotor angle	INTEGER_16	-		OSC		Tx	
0x35DF	Axis: Device data							
	1: Axis: Rated current	UNSIGNED_16	1/100		OSC		Tx	
	2: Axis: Maximum current	UNSIGNED_16	1/100		OSC		Tx	
	5: Axis: Supported feedback type	UNSIGNED_8	-				Tx	
0x35E0	Advanced settings							
	1: Current controller: Setting for identification	UNSIGNED_8	-					
	2: Sensorless synchronous control: Signal for test mode	UNSIGNED_8	-					
	3: Resolver: Position detection - dynamics	UNSIGNED_16	-					
Axis A - CiA402 profile specific area								
0x603F	Error code	UNSIGNED_16	-		OSC		Tx	
0x6040	Controlword	UNSIGNED_16	-		OSC	Rx		
0x6041	Statusword	UNSIGNED_16	-		OSC		Tx	
0x6042	vl target velocity	INTEGER_16	-		OSC	Rx		
0x6043	vl velocity demand	INTEGER_16	-		OSC		Tx	
0x6044	vl velocity actual value	INTEGER_16	-		OSC		Tx	
0x6046	vl velocity min max amount							
	1: Speed: Minimum vl	UNSIGNED_32	-		OSC	Rx		
	2: Speed: Maximum vl	UNSIGNED_32	-		OSC	Rx		
0x6048	vl velocity acceleration							
	1: Ramp: Speed interval	UNSIGNED_32	-	P	OSC	Rx		
	2: Ramp: Time interval	UNSIGNED_16	-	P	OSC	Rx		
Greyed out = read only								

Index	Name	Data type	Scaling	Attributes				
0x6049	vl velocity deceleration							
	1: Ramp: Speed interval	UNSIGNED_32	-	P	OSC	Rx		
	2: Ramp: Time interval	UNSIGNED_16	-	P	OSC	Rx		
0x605A	Quick stop option code	INTEGER_16	-	P				
0x605E	Setting/response in the event of an error	INTEGER_16	-	P				
0x6060	Modes of operation	INTEGER_8	-	P	OSC	Rx		
0x6061	Modes of operation display	INTEGER_8	-		OSC		Tx	
0x6062	Position demand value	INTEGER_32	-		OSC			
0x6063	Position actual internal value	INTEGER_32	-		OSC		Tx	
0x6064	Position actual value	INTEGER_32	-		OSC		Tx	
0x6065	Following error window	UNSIGNED_32	-	P	OSC	Rx		
0x6066	Following error time out	UNSIGNED_16	-	P	OSC	Rx		
0x6067	Position window	UNSIGNED_32	-	P				
0x6068	Position window time	UNSIGNED_16	-	P				
0x606C	Velocity actual value	INTEGER_32	$480000/2^{31}$		OSC		Tx	
0x6071	Target torque	INTEGER_16	1/10		OSC	Rx		
0x6072	Max torque	UNSIGNED_16	1/10	P	OSC	Rx		
0x6073	Max current	UNSIGNED_16	1/10	P				
0x6074	Target torque	INTEGER_16	1/10		OSC			
0x6075	Motor rated current	UNSIGNED_32	1/1000	P				CINH
0x6076	Motor rated torque	UNSIGNED_32	1/1000	P				CINH
0x6077	Torque actual value	INTEGER_16	1/10		OSC		Tx	
0x6078	Current actual value	INTEGER_16	1/10		OSC		Tx	
0x6079	DC bus: Actual voltage	UNSIGNED_32	1/1000		OSC		Tx	
0x607A	Position demand value	INTEGER_32	-		OSC	Rx		
0x607E	Polarity	UNSIGNED_8	-	P				CINH
0x6080	Max motor speed	UNSIGNED_32	-	P	OSC	Rx		
0x6085	Quick stop deceleration	UNSIGNED_32	-	P				
0x608F	Position encoder resolution							
	1: Encoder increments	UNSIGNED_32	-	P				CINH
	2: Motor revolutions	UNSIGNED_32	-	P				CINH
0x6090	Velocity encoder resolution							
	1: Encoder increments per second	UNSIGNED_32	-	P				CINH
	2: Motor revolutions per second	UNSIGNED_32	-	P				CINH
0x60B1	Velocity offset	INTEGER_32	$480000/2^{31}$	P	OSC	Rx		
0x60B2	Torque offset	INTEGER_16	1/10	P	OSC	Rx		
0x60B8	Touch probe function	UNSIGNED_16	-	P	OSC	Rx		
0x60B9	Touch probe status	UNSIGNED_16	-		OSC		Tx	
0x60BA	Touch probe pos1 pos value	INTEGER_32	-		OSC		Tx	
0x60BB	Touch probe pos1 neg value	INTEGER_32	-		OSC		Tx	
0x60BC	Touch probe pos2 pos value	INTEGER_32	-		OSC		Tx	
0x60BD	Touch probe pos2 neg value	INTEGER_32	-		OSC		Tx	
0x60C0	Interpolation sub mode select	INTEGER_16	-	P		Rx		
0x60C2	Interpolation time period							
	1: Interpolation time period value	UNSIGNED_8	-	P				
	2: Interpolation time index	INTEGER_8	-	P				
0x60E0	Positive torque limit value	UNSIGNED_16	1/10	P	OSC	Rx		
0x60E1	Negative torque limit value	UNSIGNED_16	1/10	P	OSC	Rx		
0x60F4	Following error actual value	INTEGER_32	-		OSC		Tx	
0x60FA	Control effort	INTEGER_32	$480000/2^{31}$		OSC		Tx	
0x60FC	Position demand internal value	INTEGER_32	-		OSC			
Greyed out = read only								

Index	Name	Data type	Scaling	Attributes				
0x60FD	Digital inputs	UNSIGNED_32	-		OSC		Tx	
0x60FF	Target velocity	INTEGER_32	$480000/2^{31}$		OSC	Rx		
0x6404	Motor manufacturer	STRING(50)	-	P				
0x6502	Supported drive modes	UNSIGNED_32	-					
0x67FF	ECAT: Device Profile Number	UNSIGNED_32	-					
Axis B - CiA402 profile specific area								
0x683F	Error code	UNSIGNED_16	-		OSC		Tx	
0x6840	Controlword	UNSIGNED_16	-		OSC	Rx		
0x6841	Statusword	UNSIGNED_16	-		OSC		Tx	
0x6842	vl target velocity	INTEGER_16	-		OSC	Rx		
0x6843	vl velocity demand	INTEGER_16	-		OSC		Tx	
0x6844	vl velocity actual value	INTEGER_16	-		OSC		Tx	
0x6846	vl velocity min max amount							
	1: Speed: Minimum vl	UNSIGNED_32	-		OSC	Rx		
	2: Speed: Maximum vl	UNSIGNED_32	-		OSC	Rx		
0x6848	vl velocity acceleration							
	1: Ramp: Speed interval	UNSIGNED_32	-	P	OSC	Rx		
	2: Ramp: Time interval	UNSIGNED_16	-	P	OSC	Rx		
0x6849	vl velocity deceleration							
	1: Ramp: Speed interval	UNSIGNED_32	-	P	OSC	Rx		
	2: Ramp: Time interval	UNSIGNED_16	-	P	OSC	Rx		
0x685A	Quick stop option code	INTEGER_16	-	P				
0x685E	Setting/response in the event of an error	INTEGER_16	-	P				
0x6860	Modes of operation	INTEGER_8	-	P	OSC	Rx		
0x6861	Modes of operation display	INTEGER_8	-		OSC		Tx	
0x6862	Position demand value	INTEGER_32	-		OSC			
0x6863	Position actual internal value	INTEGER_32	-		OSC		Tx	
0x6864	Position actual value	INTEGER_32	-		OSC		Tx	
0x6865	Following error window	UNSIGNED_32	-	P	OSC	Rx		
0x6866	Following error time out	UNSIGNED_16	-	P	OSC	Rx		
0x6867	Position window	UNSIGNED_32	-	P				
0x6868	Position window time	UNSIGNED_16	-	P				
0x686C	Velocity actual value	INTEGER_32	$480000/2^{31}$		OSC		Tx	
0x6871	Target torque	INTEGER_16	1/10		OSC	Rx		
0x6872	Max torque	UNSIGNED_16	1/10	P	OSC	Rx		
0x6873	Max current	UNSIGNED_16	1/10	P				
0x6874	Target torque	INTEGER_16	1/10		OSC			
0x6875	Motor rated current	UNSIGNED_32	1/1000	P				CINH
0x6876	Motor rated torque	UNSIGNED_32	1/1000	P				CINH
0x6877	Torque actual value	INTEGER_16	1/10		OSC		Tx	
0x6878	Current actual value	INTEGER_16	1/10		OSC		Tx	
0x6879	DC bus: Actual voltage	UNSIGNED_32	1/1000		OSC		Tx	
0x687A	Position demand value	INTEGER_32	-		OSC	Rx		
0x687E	Polarity	UNSIGNED_8	-	P				CINH
0x6880	Max motor speed	UNSIGNED_32	-	P	OSC	Rx		
0x6885	Quick stop deceleration	UNSIGNED_32	-	P				
0x688F	Position encoder resolution							
	1: Encoder increments	UNSIGNED_32	-	P				CINH
	2: Motor revolutions	UNSIGNED_32	-	P				CINH
Greyed out = read only								

Index	Name	Data type	Scaling	Attributes				
0x6890	Velocity encoder resolution							
	1: Encoder increments per second	UNSIGNED_32	-	P				CINH
	2: Motor revolutions per second	UNSIGNED_32	-	P				CINH
0x68B1	Velocity offset	INTEGER_32	$480000/2^{31}$	P	OSC	Rx		
0x68B2	Torque offset	INTEGER_16	1/10	P	OSC	Rx		
0x68B8	Touch probe function	UNSIGNED_16	-	P	OSC	Rx		
0x68B9	Touch probe status	UNSIGNED_16	-		OSC		Tx	
0x68BA	Touch probe pos1 pos value	INTEGER_32	-		OSC		Tx	
0x68BB	Touch probe pos1 neg value	INTEGER_32	-		OSC		Tx	
0x68BC	Touch probe pos2 pos value	INTEGER_32	-		OSC		Tx	
0x68BD	Touch probe pos2 neg value	INTEGER_32	-		OSC		Tx	
0x68C0	Interpolation sub mode select	INTEGER_16	-	P		Rx		
0x68C2	Interpolation time period							
	1: Interpolation time period value	UNSIGNED_8	-	P				
	2: Interpolation time index	INTEGER_8	-	P				
0x68E0	Positive torque limit value	UNSIGNED_16	1/10	P	OSC	Rx		
0x68E1	Negative torque limit value	UNSIGNED_16	1/10	P	OSC	Rx		
0x68F4	Following error actual value	INTEGER_32	-		OSC		Tx	
0x68FA	Control effort	INTEGER_32	$480000/2^{31}$		OSC		Tx	
0x68FC	Position demand internal value	INTEGER_32	-		OSC			
0x68FD	Digital inputs	UNSIGNED_32	-		OSC		Tx	
0x68FF	Target velocity	INTEGER_32	$480000/2^{31}$		OSC	Rx		
0x6C04	Motor manufacturer	STRING(50)	-	P				
0x6D02	Supported drive modes	UNSIGNED_32	-					
0x6FFF	ECAT: Device Profile Number	UNSIGNED_32	-					
Greyed out = read only								

11.2 Structure of the parameter set file

The parameter set file is designed as an extendable file. This means that it is possible to add header components later on without interrupting previous algorithms. Therefore the file is divided into three sections:

1. File header
   - The file header contains important information on the file as a whole, including its size, the checksum, the file header type, the number of data headers, and their position from the start of the file.
2. Data header
   - The data header contains information on a specific part of the data within the data area. These are for instance the data type and data interpretation and thus the type of data header, the data offset in the file, and the size of the data packet.
3. Data
   - The data is a densely packed arrangement of data, the sense of which only becomes clear when the type of data header specified is known.

11.2.1 File header

Byte offset

0	4	8	12	16	20	24	28	32
Type	Version	Size	Manufacturer	Product	Version	Data start	Data size	CRC

<table>
<tr><th>Byte offset</th><th>Meaning</th><th>Info</th></tr>
<tr><td>0</td><td>File type</td><td>File identification
Currently only the following values are defined:
• 2: Storage parameter set (the file exclusively contains parameters provided for persistence).
• 3: Complete parameter set (the file contains all parameters).</td></tr>
<tr><td>4</td><td>File version</td><td>Version number of the file</td></tr>
<tr><td>8</td><td>File size</td><td>Total size of the file in [bytes]</td></tr>
<tr><td>12</td><td>Manufacturer's identification mark according to object directory</td><td rowspan="3">Identification of the device which this parameter set file belongs to. If the values do not comply with those of the object directory during the import, an attempt to execute the import is nevertheless carried out. Parameters which could not be imported can be identified later by means of the parameter set error report.</td></tr>
<tr><td>16</td><td>Product ID according to object directory</td></tr>
<tr><td>20</td><td>Version identifier of the object directory</td></tr>
<tr><td>24</td><td>Start of the data description area</td><td>Offset of the data description area in [bytes]
(offset from the start of the file).</td></tr>
<tr><td>28</td><td>Size of the data description area</td><td>Size of the data description area in [bytes]</td></tr>
<tr><td>32</td><td>CRC32 (as defined in IEEE 802.3)</td><td>Cyclic Redundancy Check (CRC32) across the entire file; 0 must be entered in the checksum field at the time of calculation.
▸ Cyclic redundancy check (CRC) - parameter set comparison on the basis of the checksum (📖 21)</td></tr>
</table>

11.2.2 Data header

Byte offset

	0	4	8	
...	Type	Data start	Data size	...

Byte offset	Meaning	Info
0	Type of the data area	Identification of the data area Currently only the following values are defined: • 1: The data area contains parameter values. ▶ "PAR_ParamHeader" header type • 2: The data area contains error messages of the last import. ▶ "PAR_ParamErrorHeader" header type
4	Start of the data area	Offset of the data area in [bytes] (offset from the start of the file).
8	Size of the data area	Size of the data area in [bytes]

"PAR_ParamHeader" header type

If the identification "1" is given for the data area type, the parameter values are successively arranged in the following layout within the data area:

Byte offset

	0	2	4	6	
...	Data type size	Index	Subindex	Value	...

Byte offset	Meaning	Info
0	Data type size of the parameter	Number of bytes which are assigned by the parameter value.
2	Index of the parameter object	Index of the parameter in the object directory which the following parameter value belongs to.
4	Subindex of the parameter object	Subindex of the parameter in the object directory which the following parameter value belongs to.
6	Parameter value	The parameter value assigns the number of data bytes which are listed under "Data type size of the parameter". At least, however, always one byte.

"PAR_ParamErrorHeader" header type

If the identification "2" is given for the data area type, the error messages are successively arranged in the following layout within the data area:

Byte offset

	0	4	6	
...	Error code	Index	Subindex	...

Byte offset	Meaning	Info
0	Error code in accordance with CiA301 or EtherCAT	SDO abort code which was generated when the object with the following index/subindex combination was accessed. ▶ SDO abort codes (📖 27)
4	Index of the parameter object	Index of the parameter in the object directory which the error code belongs to.
6	Subindex of the parameter object	Subindex of the parameter in the object directory which the error code belongs to.

11.3 Communication objects

This chapter describes the "invisible" communication objects of the i700 servo inverter. This information is important for the integration of the controller with an external control.

Objects described in this chapter

All objects are of "RECORD" data type.

Object Axis A	 Axis B	Name
PDO mapping		
0x1600	0x1610	RPDO-->Axis A/B: Cyclic sync position mode (csp)
0x1601	0x1611	RPDO-->Axis A/B: Cyclic sync torque mode (cst)
0x1602	0x1612	RPDO-->Axis A/B: Cyclic sync velocity mode (csv)
0x1603	0x1613	RPDO-->Axis A/B: Velocity mode (vl)
0x1604	0x1614	RPDO-->Axis A/B: Touch probe (TP)
0x1605	0x1615	RPDO-->Axis A/B: Freely configurable (user)
0x1606	0x1616	RPDO-->Axis A/B: Torque limits
0x1607	0x1617	RPDO-->Axis A/B: Speed limit values
0x1A00	0x1A10	Axis A/B-->TPDO: Cyclic sync position mode (csp)
0x1A01	0x1A11	Axis A/B-->TPDO: Cyclic sync torque mode (cst)
0x1A02	0x1A12	Axis A/B-->TPDO: Cyclic sync velocity mode (csv)
0x1A03	0x1A13	Axis A/B-->TPDO: Velocity mode (vl)
0x1A04	0x1A14	Axis A/B-->TPDO: Touch probe (TP)
0x1A05	0x1A15	Axis A/B-->TPDO: Freely configurable (user)
0x1A06	0x1A16	Axis A/B-->TPDO: Additional status information
Sync manager		
0x1C00		Sync Manager: Communication type
0x1C12		Sync Manager 2 (RPDO-->Device): PDO mapping
0x1C13		Sync Manager 3 (RPDO-->Device): PDO mapping
0x1C32		Sync Manager 2 (RPDO-->Device): Parameter
0x1C33		Sync Manager 3 (Device-->TPDO): Parameter

0x1600 - RPDO-->Axis A: Cyclic sync position mode (csp)

Fixed, preconfigured PDO mapping object for "Cyclic sync position mode"

Sub.	Name	Lenze setting	Data type
1	(csp) RPDO-->A: 0x6040 (Controlword)	0x60400010	UNSIGNED_32
2	(csp) RPDO-->A: 0x2830 (Lenze control word)	0x28300010	UNSIGNED_32
3	(csp) RPDO-->A: 0x6060 (Modes of operation)	0x60600008	UNSIGNED_32
4	(csp) RPDO-->A: 0x60B2 (Torque: Offset)	0x60B20010	UNSIGNED_32
5	(csp) RPDO-->A: 0x607A (Target position)	0x607A0020	UNSIGNED_32
6	(csp) RPDO-->A: 0x60B1 (Velocity offset)	0x60B10020	UNSIGNED_32
7	(csp) RPDO-->A: 0x2902 (speed controller - load starting value)	0x29020010	UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1601 - RPDO-->Axis A: Cyclic sync torque mode (cst)

Fixed, preconfigured PDO mapping object for "Cyclic sync torque mode"

Sub.	Name	Lenze setting	Data type
1	(cst) RPDO-->A: 0x6040 (Controlword)	0x60400010	UNSIGNED_32
2	(cst) RPDO-->A: 0x2830 (Lenze control word)	0x28300010	UNSIGNED_32
3	(cst) RPDO-->A: 0x6060 (Modes of operation)	0x60600008	UNSIGNED_32
4	(cst) RPDO-->A: 0x60B2 (Torque: Offset)	0x60B20010	UNSIGNED_32
5	(cst) RPDO-->A: 0x6071 (Target torque)	0x60710010	UNSIGNED_32
6	(cst) RPDO-->A: 0x2946/1 (upper speed limit)	0x29460120	UNSIGNED_32
7	(cst) RPDO-->A: 0x2946/2 (lower speed limit)	0x29460220	UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1602 - RPDO-->Axis A: Cyclic sync velocity mode (csv)

Fixed, preconfigured PDO mapping object for "Cyclic sync velocity mode"

Sub.	Name	Lenze setting	Data type
1	(csv) RPDO-->A: 0x6040 (Controlword)	0x60400010	UNSIGNED_32
2	(csv) RPDO-->A: 0x2830 (Lenze control word)	0x28300010	UNSIGNED_32
3	(csv) RPDO-->A: 0x6060 (Modes of operation)	0x60600008	UNSIGNED_32
4	(csv) RPDO-->A: 0x60B2 (Torque: Offset)	0x60B20010	UNSIGNED_32
5	(csv) RPDO-->A: 0x60FF (Target velocity)	0x60FF0020	UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1603 - RPDO-->Axis A: Velocity mode (vl)

Fixed, preconfigured PDO mapping object for "Velocity mode"

Sub.	Name	Lenze setting	Data type
1	(vl) RPDO-->A: 0x6040 (Controlword)	0x60400010	UNSIGNED_32
2	(vl) RPDO-->A: 0x2830 (Lenze control word)	0x28300010	UNSIGNED_32
3	(vl) RPDO-->A: 0x6060 (Modes of operation)	0x60600008	UNSIGNED_32
4	(vl) RPDO-->A: 0x6042 (vl target velocity)	0x60420010	UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1604 - RPDO-->Axis A: Touch probe (TP)

Fixed, preconfigured PDO mapping object for touch probe detection

Sub.	Name	Lenze setting	Data type
1	(TP) RPDO-->A: 0x60B8 (Touch probe function)	0x60B80010	UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1605 - RPDO-->Axis A: Freely configurable (user)

PDO mapping object freely configurable by the user for process data from the controller to the servo inverter

Sub.	Name	Lenze setting	Data type
1	(user) RPDO-->A: xxxx freely configurable object 1	0x00000000	UNSIGNED_32
2	(user) RPDO-->A: xxxx freely configurable object 2	0x00000000	UNSIGNED_32
3	(user) RPDO-->A: xxxx freely configurable object 3	0x00000000	UNSIGNED_32
4	(user) RPDO-->A: xxxx freely configurable object 4	0x00000000	UNSIGNED_32
5	(user) RPDO-->A: xxxx freely configurable object 5	0x00000000	UNSIGNED_32
6	(user) RPDO-->A: xxxx freely configurable object 6	0x00000000	UNSIGNED_32
7	(user) RPDO-->A: xxxx freely configurable object 7	0x00000000	UNSIGNED_32
8	(user) RPDO-->A: xxxx freely configurable object 8	0x00000000	UNSIGNED_32
☑ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1606 - RPDO-->Axis A: Torque limits

Fixed, preconfigured PDO mapping object for torque limits

Sub.	Name	Lenze setting	Data type
1	(torque) RPDO-->A: 0x60E0 (Positive torque limit value)	0x60E00010	UNSIGNED_32
2	(torque) RPDO-->A: 0x60E1 (Negative torque limit value)	0x60E10010	UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1607 - RPDO --> Axis A: Speed Limits

Sub.	Name	Lenze setting	Data type
1	(Speed) RPDO-->A: 0x2946/1 (upper speed limit)	692453664	UNSIGNED_32
2	(Speed) RPDO-->A: 0x2946/2 Lower speed limit	692453920	UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1610 - RPDO-->Axis B: Cyclic sync position mode (csp)

Fixed, preconfigured PDO mapping object for "Cyclic sync position mode"

Sub.	Name	Lenze setting	Data type
1	(csp) RPDO-->B: 0x6840 (Controlword)	0x68400010	UNSIGNED_32
2	(csp) RPDO-->B: 0x3030 (Lenze control word)	0x30300010	UNSIGNED_32
3	(csp) RPDO-->B: 0x6860 (Modes of operation)	0x68600008	UNSIGNED_32
4	(csp) RPDO-->B: 0x68B2 (Torque: Offset)	0x68B20010	UNSIGNED_32
5	(csp) RPDO-->B: 0x687A (Target position)	0x687A0020	UNSIGNED_32
6	(csp) RPDO-->B: 0x68B1 (Velocity offset)	0x68B10020	UNSIGNED_32
7	(csp) RPDO-->B: 0x3102 speed controller - load starting value	0x31020010	UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1611 - RPDO-->Axis B: Cyclic sync torque mode (cst)

Fixed, preconfigured PDO mapping object for "Cyclic sync torque mode"

Sub.	Name	Lenze setting	Data type
1	(cst) RPDO-->B: 0x6840 (Controlword)	0x68400010	UNSIGNED_32
2	(cst) RPDO-->B: 0x3030 (Lenze control word)	0x30300010	UNSIGNED_32
3	(cst) RPDO-->B: 0x6860 (Modes of operation)	0x68600008	UNSIGNED_32
4	(cst) RPDO-->B: 0x68B2 (Torque: Offset)	0x68B20010	UNSIGNED_32
5	(cst) RPDO-->B: 0x6871 (Target torque)	0x68710010	UNSIGNED_32
6	(cst) RPDO-->B: 0x3146/1 upper speed limit	0x31460120	UNSIGNED_32
7	(cst) RPDO-->B: 0x3146/2 lower speed limit	0x31460220	UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1612 - RPDO-->Axis B: Cyclic sync velocity mode (csv)

Fixed, preconfigured PDO mapping object for "Cyclic sync velocity mode"

Sub.	Name	Lenze setting	Data type
1	(csv) RPDO-->B: 0x6840 (Controlword)	0x68400010	UNSIGNED_32
2	(csv) RPDO-->B: 0x3030 (Lenze control word)	0x30300010	UNSIGNED_32
3	(csv) RPDO-->B: 0x6860 (Modes of operation)	0x68600008	UNSIGNED_32
4	(csv) RPDO-->B: 0x68B2 (Torque: Offset)	0x68B20010	UNSIGNED_32
5	(csv) RPDO-->B: 0x68FF (Target velocity)	0x68FF0020	UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1613 - RPDO-->Axis B: Velocity mode (vl)

Fixed, preconfigured PDO mapping object for "Velocity mode"

Sub.	Name	Lenze setting	Data type
1	(vl) RPDO-->B: 0x6840 (Controlword)	0x68400010	UNSIGNED_32
2	(vl) RPDO-->B: 0x3030 (Lenze control word)	0x30300010	UNSIGNED_32
3	(vl) RPDO-->B: 0x6860 (Modes of operation)	0x68600008	UNSIGNED_32
4	(vl) RPDO-->B: 0x6842 (vl target velocity)	0x68420010	UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1614 - RPDO-->Axis B: Touch probe (TP)

Fixed, preconfigured PDO mapping object for touch probe detection

Sub.	Name	Lenze setting	Data type
1	(TP) RPDO-->B: 0x68B8 (Touch probe function)	0x68B80010	UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1615 - RPDO-->Axis B: Freely configurable (user)

PDO mapping object freely configurable by the user for process data from the controller to the servo inverter i700

Sub.	Name	Lenze setting	Data type
1	(user) RPDO-->B: xxxx freely configurable object 1	0x00000000	UNSIGNED_32
2	(user) RPDO-->B: xxxx freely configurable object 2	0x00000000	UNSIGNED_32
3	(user) RPDO-->B: xxxx freely configurable object 3	0x00000000	UNSIGNED_32
4	(user) RPDO-->B: xxxx freely configurable object 4	0x00000000	UNSIGNED_32
5	(user) RPDO-->B: xxxx freely configurable object 5	0x00000000	UNSIGNED_32
6	(user) RPDO-->B: xxxx freely configurable object 6	0x00000000	UNSIGNED_32
7	(user) RPDO-->B: xxxx freely configurable object 7	0x00000000	UNSIGNED_32
8	(user) RPDO-->B: xxxx freely configurable object 8	0x00000000	UNSIGNED_32
☑ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1616 - RPDO-->Axis B: Torque limits

Fixed, preconfigured PDO mapping object for torque limits

Sub.	Name	Lenze setting	Data type
1	(torque) RPDO-->B: 0x68E0 (Positive torque limit value)	0x68E00010	UNSIGNED_32
2	(torque) RPDO-->B: 0x68E1 (Negative torque limit value)	0x68E10010	UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1617 - RPDO --> Axis B: Speed limits

Sub.	Name	Lenze setting	Data type
1	(Speed) RPDO-->B: 0x3146/1 (upper speed limit)	826671392	UNSIGNED_32
2	(Speed) RPDO-->B: 0x3146/2 (Lower speed limit)	826671648	UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1A00 - Axis A-->TPDO: Cyclic sync position mode (csp)

Fixed, preconfigured PDO mapping object for "Cyclic sync position mode"

Sub.	Name	Lenze setting	Data type
1	(csp) A-->TPDO: 0x6041 (Statusword)	0x60410010	UNSIGNED_32
2	(csp) A-->TPDO: 0x2831 (Lenze status word)	0x28310010	UNSIGNED_32
3	(csp) A-->TPDO: 0x6061 (Modes of operation display)	0x60610008	UNSIGNED_32
4	(csp) A-->TPDO: 0x603F (Error code)	0x603F0010	UNSIGNED_32
5	(csp) A-->TPDO: 0x606C (Velocity actual value)	0x606C0020	UNSIGNED_32
6	(csp) A-->TPDO: 0x6077 (Torque actual value)	0x60770010	UNSIGNED_32
7	(csp) A-->TPDO: 0x6064 (Position actual value)	0x60640020	UNSIGNED_32
8	(csp) A-->TPDO: 0x60F4 (Following error actual value)	0x60F40020	UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1A01 - Axis A-->TPDO: Cyclic sync torque mode (cst)

Fixed, preconfigured PDO mapping object for "Cyclic sync torque mode"

Sub.	Name	Lenze setting	Data type
1	(cst) A-->TPDO: 0x6041 (Statusword)	0x60410010	UNSIGNED_32
2	(cst) A-->TPDO: 0x2831 (Lenze status word)	0x28310010	UNSIGNED_32
3	(cst) A-->TPDO: 0x6061 (Modes of operation display)	0x60610008	UNSIGNED_32
4	(cst) A-->TPDO: 0x603F (Error code)	0x603F0010	UNSIGNED_32
5	(cst) A-->TPDO: 0x606C (Velocity actual value)	0x606C0020	UNSIGNED_32
6	(cst) A-->TPDO: 0x6077 (Torque actual value)	0x60770010	UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1A02 - Axis A-->TPDO: Cyclic sync velocity mode (csv)

Fixed, preconfigured PDO mapping object for "Cyclic sync velocity mode"

Sub.	Name	Lenze setting	Data type
1	(csv) A-->TPDO: 0x6041 (Statusword)	0x60410010	UNSIGNED_32
2	(csv) A-->TPDO: 0x2831 (Lenze status word)	0x28310010	UNSIGNED_32
3	(csv) A-->TPDO: 0x6061 (Modes of operation display)	0x60610008	UNSIGNED_32
4	(csv) A-->TPDO: 0x603F (Error code)	0x603F0010	UNSIGNED_32
5	(csv) A-->TPDO: 0x606C (Velocity actual value)	0x606C0020	UNSIGNED_32
6	(csv) A-->TPDO: 0x6077 (Torque actual value)	0x60770010	UNSIGNED_32
7	(csv) A-->TPDO: 0x6064 (Position actual value)	0x60640020	UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1A03 - Axis A-->TPDO: Velocity mode (vl)

Fixed, preconfigured PDO mapping object for "Velocity mode"

Sub.	Name	Lenze setting	Data type
1	(vl) A-->TPDO: 0x6041 (Statusword)	0x60410010	UNSIGNED_32
2	(vl) A-->TPDO: 0x2831 (Lenze status word)	0x28310010	UNSIGNED_32
3	(vl) A-->TPDO: 0x6061 (Modes of operation display)	0x60610008	UNSIGNED_32
4	(vl) A-->TPDO: 0x603F (Error code)	0x603F0010	UNSIGNED_32
5	(vl) A-->TPDO: 0x6044 (vl velocity actual value)	0x60440010	UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1A04 - Axis A-->TPDO: Touch probe (TP)

Fixed, preconfigured PDO mapping object for touch probe detection

Sub.	Name	Lenze setting	Data type
1	(TP) A-->TPDO: 0x60B9 (Touch probe status)	0x60B90010	UNSIGNED_32
2	(TP) A-->TPDO: 0x60BA (Touch probe pos1 pos value)	0x60BA0020	UNSIGNED_32
3	(TP) A-->TPDO: 0x60BB (Touch probe pos1 neg value)	0x60BB0020	UNSIGNED_32
4	(TP) A-->TPDO: 0x60BC (Touch probe pos2 pos value)	0x60BC0020	UNSIGNED_32
5	(TP) A-->TPDO: 0x60BD (Touch probe pos2 neg value)	0x60BD0020	UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1A05 - Axis A-->TPDO: Freely configurable (user)

PDO mapping object freely configurable by the user for process data from the i700 servo inverter to the controller

Sub.	Name	Lenze setting	Data type
1	(user) A-->TPDO: xxxx freely configurable object 1	0x00000000	UNSIGNED_32
2	(user) A-->TPDO: xxxx freely configurable object 2	0x00000000	UNSIGNED_32
3	(user) A-->TPDO: xxxx freely configurable object 3	0x00000000	UNSIGNED_32
4	(user) A-->TPDO: xxxx freely configurable object 4	0x00000000	UNSIGNED_32
5	(user) A-->TPDO: xxxx freely configurable object 5	0x00000000	UNSIGNED_32
6	(user) A-->TPDO: xxxx freely configurable object 6	0x00000000	UNSIGNED_32
7	(user) A-->TPDO: xxxx freely configurable object 7	0x00000000	UNSIGNED_32
8	(user) A-->TPDO: xxxx freely configurable object 8	0x00000000	UNSIGNED_32
☑ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1A06 - Axis A-->TPDO: Additional status information

Sub.	Name	Lenze setting	Data type
1	(State) A-->TPDO: 0x2833 Lenze Statusword 2	674430992	UNSIGNED_32
2	(State) A-->TPDO: 0x60FD Digital inputs	1627193376	UNSIGNED_32
3	(State) A-->TPDO: xxxx Free config. object 1	0	UNSIGNED_32
4	(State) A-->TPDO: xxxx Free config. object 2	0	UNSIGNED_32
5	(State) A-->TPDO: xxxx Free config. object 3	0	UNSIGNED_32
6	(State) A-->TPDO: xxxx Free config. object 4	0	UNSIGNED_32
7	(State) A-->TPDO: xxxx Free config. object 5	0	UNSIGNED_32
8	(State) A-->TPDO: xxxx Free config. object 6	0	UNSIGNED_32
☑ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1A10 - Axis B-->TPDO: Cyclic sync position mode (csp)

Fixed, preconfigured PDO mapping object for "Cyclic sync position mode"

Sub.	Name	Lenze setting	Data type
1	(csp) B-->TPDO: 0x6841 (Statusword)	0x68410010	UNSIGNED_32
2	(csp) B-->TPDO: 0x3031 (Lenze status word)	0x30310010	UNSIGNED_32
3	(csp) B-->TPDO: 0x6861 (Modes of operation display)	0x68610008	UNSIGNED_32
4	(csp) B-->TPDO: 0x683F (Error code)	0x683F0010	UNSIGNED_32
5	(csp) B-->TPDO: 0x686C (Velocity actual value)	0x686C0020	UNSIGNED_32
6	(csp) B-->TPDO: 0x6877 (Torque actual value)	0x68770010	UNSIGNED_32
7	(csp) B-->TPDO: 0x6864 (Position actual value)	0x68640020	UNSIGNED_32
8	(csp) B-->TPDO: 0x68F4 (Following error actual value)	0x68F40020	UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1A11 - Axis B-->TPDO: Cyclic sync torque mode (cst)

Fixed, preconfigured PDO mapping object for "Cyclic sync torque mode"

Sub.	Name	Lenze setting	Data type
1	(cst) B-->TPDO: 0x6841 (Statusword)	0x68410010	UNSIGNED_32
2	(cst) B-->TPDO: 0x3031 (Lenze status word)	0x30310010	UNSIGNED_32
3	(cst) B-->TPDO: 0x6861 (Modes of operation display)	0x68610008	UNSIGNED_32
4	(cst) B-->TPDO: 0x683F (Error code)	0x683F0010	UNSIGNED_32
5	(cst) B-->TPDO: 0x686C (Velocity actual value)	0x686C0020	UNSIGNED_32
6	(cst) B-->TPDO: 0x6877 (Torque actual value)	0x68770010	UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1A12 - Axis B-->TPDO: Cyclic sync velocity mode (csv)

Fixed, preconfigured PDO mapping object for "Cyclic sync velocity mode"

Sub.	Name	Lenze setting	Data type
1	(csv) B-->TPDO: 0x6841 (Statusword)	0x68410010	UNSIGNED_32
2	(csv) B-->TPDO: 0x3031 (Lenze status word)	0x30310010	UNSIGNED_32
3	(csv) B-->TPDO: 0x6861 (Modes of operation display)	0x68610008	UNSIGNED_32
4	(csv) B-->TPDO: 0x683F (Error code)	0x683F0010	UNSIGNED_32
5	(csv) B-->TPDO: 0x686C (Velocity actual value)	0x686C0020	UNSIGNED_32
6	(csv) B-->TPDO: 0x6877 (Torque actual value)	0x68770010	UNSIGNED_32
7	(csv) B-->TPDO: 0x6864 (Position actual value)	0x68640020	UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1A13 - Axis B-->TPDO: Velocity mode (vl)

Fixed, preconfigured PDO mapping object for "Velocity mode"

Sub.	Name	Lenze setting	Data type
1	(vl) B-->TPDO: 0x6841 (Statusword)	0x68410010	UNSIGNED_32
2	(vl) B-->TPDO: 0x3031 (Lenze status word)	0x30310010	UNSIGNED_32
3	(vl) B-->TPDO: 0x6861 (Modes of operation display)	0x68610008	UNSIGNED_32
4	(vl) B-->TPDO: 0x683F (Error code)	0x683F0010	UNSIGNED_32
5	(vl) B-->TPDO: 0x6844 (vl velocity actual value)	0x68440010	UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1A14 - Axis B-->TPDO: Touch probe (TP)

Fixed, preconfigured PDO mapping object for touch probe detection

Sub.	Name	Lenze setting	Data type
1	(TP) B-->TPDO: 0x68B9 (Touch probe status)	0x68B90010	UNSIGNED_32
2	(TP) B-->TPDO: 0x68BA (Touch probe pos1 pos value)	0x68BA0020	UNSIGNED_32
3	(TP) B-->TPDO: 0x68BB (Touch probe pos1 neg value)	0x68BB0020	UNSIGNED_32
4	(TP) B-->TPDO: 0x68BC (Touch probe pos2 pos value)	0x68BC0020	UNSIGNED_32
5	(TP) B-->TPDO: 0x68BD (Touch probe pos2 neg value)	0x68BD0020	UNSIGNED_32
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1A15 - Axis B-->TPDO: Freely configurable (user)

PDO mapping object freely configurable by the user for process data from the i700 servo inverter to the controller

Sub.	Name	Lenze setting	Data type
1	(user) B-->TPDO: xxxx freely configurable object 1	0x00000000	UNSIGNED_32
2	(user) B-->TPDO: xxxx freely configurable object 2	0x00000000	UNSIGNED_32
3	(user) B-->TPDO: xxxx freely configurable object 3	0x00000000	UNSIGNED_32
4	(user) B-->TPDO: xxxx freely configurable object 4	0x00000000	UNSIGNED_32
5	(user) B-->TPDO: xxxx freely configurable object 5	0x00000000	UNSIGNED_32
6	(user) B-->TPDO: xxxx freely configurable object 6	0x00000000	UNSIGNED_32
7	(user) B-->TPDO: xxxx freely configurable object 7	0x00000000	UNSIGNED_32
8	(user) B-->TPDO: xxxx freely configurable object 8	0x00000000	UNSIGNED_32
☑ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1A16 - Axis B-->TPDO: Additional status information

Sub.	Name	Lenze setting	Data type
1	(State) B-->TPDO: 0x3033 Lenze Statusword 2	808648720	UNSIGNED_32
2	(State) B-->TPDO: 0x68FD Digital inputs	1761411104	UNSIGNED_32
3	(State) B-->TPDO: xxxx Free config. object 1	0	UNSIGNED_32
4	(State) B-->TPDO: xxxx Free config. object 2	0	UNSIGNED_32
5	(State) B-->TPDO: xxxx Free config. object 3	0	UNSIGNED_32
6	(State) B-->TPDO: xxxx Free config. object 4	0	UNSIGNED_32
7	(State) B-->TPDO: xxxx Free config. object 5	0	UNSIGNED_32
8	(State) B-->TPDO: xxxx Free config. object 6	0	UNSIGNED_32
☑ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1C00 - Sync Manager: Communication type

Sub.	Name	Lenze setting	Data type
1	SM1: Communication type	1: Receive mailbox	UNSIGNED_8
2	SM2: Communication type	2: Transmit mailbox	UNSIGNED_8
3	SM3: Communication type	3: Transit process data	UNSIGNED_8
4	SM4: Communication type	4: Receive process data	UNSIGNED_8
☐ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1C12 - Sync Manager 2 (RPDO-->Device): PDO mapping

Sub.	Name	Lenze setting	Data type
1	SM2 (RPDO-->Device): Assignment PDO 1	0x1600	UNSIGNED_16
2	SM2 (RPDO-->Device): Assignment PDO 2	0x1604	UNSIGNED_16
3	SM2 (RPDO-->Device): Assignment PDO 3	0x1606	UNSIGNED_16
4	SM2 (RPDO-->Device): Assignment PDO 4	0x1607	UNSIGNED_16
5	SM2 (RPDO-->Device): Assignment PDO 5	0x1610	UNSIGNED_16
6	SM2 (RPDO-->Device): Assignment PDO 6	0x1614	UNSIGNED_16
7	SM2 (RPDO-->Device): Assignment PDO 7	0x1616	UNSIGNED_16
8	SM2 (RPDO-->Device): Assignment PDO 8	0x1617	UNSIGNED_16
9	SM2 (RPDO-->Device): PDO assignment 9	0x0000	UNSIGNED_16
10	SM2 (RPDO-->Device): PDO assignment 10	0x0000	UNSIGNED_16
☑ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1C13 - Sync Manager 3 (RPDO-->Device): PDO mapping

Sub.	Name	Lenze setting	Data type
1	SM3 (Device-->TPDO): Assignment PDO 1	0x1A00	UNSIGNED_16
2	SM3 (Device-->TPDO): Assignment PDO 2	0x1A04	UNSIGNED_16
3	SM3 (Device-->TPDO): Assignment PDO 3	0x1A06	UNSIGNED_16
4	SM3 (Device-->TPDO): Assignment PDO 4	0x1A10	UNSIGNED_16
5	SM3 (Device-->TPDO): Assignment PDO 5	0x1A14	UNSIGNED_16
6	SM3 (Device-->TPDO): Assignment PDO 6	0x1A16	UNSIGNED_16
7	SM3 (Device-->TPDO): PDO assignment 7	0x0000	UNSIGNED_16
8	SM3 (Device-->TPDO): PDO assignment 8	0x0000	UNSIGNED_16
☑ Write access ☐ CINH ☐ OSC ☐ P ☐ RX ☐ TX			

0x1C32 - Sync Manager 2 (RPDO-->Device): Parameter

Sub.	Name	Lenze setting	Data type
1	SM2 (RPDO-->Device): Sync type	0: Free Run	UNSIGNED_16
2	SM2 (RPDO-->Device): Cycle time	250000 ns	UNSIGNED_32
3	SM2 (RPDO-->Device): Shift time	0 ns	UNSIGNED_32
4	SM2 (RPDO-->Device): Supported sync types	0x0825	UNSIGNED_16
5	SM2 (RPDO-->Device): Min. cycle time	125000 ns	UNSIGNED_32
6	SM2 (RPDO-->Device): Signal processing time	0 ns	UNSIGNED_32
9	SM2 (RPDO-->Device): Delay time	0 ns	UNSIGNED_32
11	SM2 (RPDO-->Device): Counter - missing events		UNSIGNED_16
12	SM2 (RPDO-->Device): Cycle time too short		UNSIGNED_16
32	SM2 (RPDO-->Device): Sync error		BOOL

0x1C33 - Sync Manager 3 (Device-->TPDO): Parameter

Sub.	Name	Lenze setting	Data type
1	SM3 (Device-->TPDO): Sync type	0: Free Run	UNSIGNED_16
2	SM3 (Device-->TPDO): Cycle time	250000 ns	UNSIGNED_32
3	SM3 (Device-->TPDO): Shift time	62500 ns	UNSIGNED_32
4	SM3 (Device-->TPDO): Supported sync types	0x0825	UNSIGNED_16
5	SM3 (Device-->TPDO): Min. cycle time	125000 ns	UNSIGNED_32
6	SM3 (Device-->TPDO): Signal processing time	0 ns	UNSIGNED_32
9	SM3 (Device-->TPDO): Delay time	0 ns	UNSIGNED_32
11	SM3 (Device-->TPDO): Counter - missing events		UNSIGNED_16
12	SM3 (Device-->TPDO): Cycle time too short		UNSIGNED_16
32	SM3 (Device-->TPDO): Sync error		BOOL

11.4 Licences

lwIP - TCP/IP stack

Numbers

Index

FEEDBACK

Your opinion is important to us

These instructions were created to the best of our knowledge and belief to give you the best possible support for handling our product.

Perhaps we have not always succeeded in achieving this goal. If you notice this, please send us your suggestions and criticism in a short e-mail to:

feedback-docu@lenze.de

Thank you for your support.

Your Lenze documentation team

i700 servo inverter · Reference manual · EDSE70ACMxx · 13410788 · DMS 1.5 EN · 03/2014 · TD05

Lenze Automation GmbH
Hans-Lenze-Str. 1
D-31855 Aerzen
Germany
+49 5154 82-0
lenze@lenze.com
www.lenze.com

Service

Lenze Service GmbH
Breslauer Straße 3
D-32699 Extertal
Germany
008000 24 46877 (24 h helpline)
+49 5154 82-1112
service@lenze.com